(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105677298 A
(43)申请公布日 2016.06.15

(21)申请号 201511028580.6

(22)申请日 2015.12.30

(71)申请人 李朝波
地址 511400 广东省广州市番禺区洛溪新城华荟明苑华茵路53栋602号

(72)发明人 李朝波

(74)专利代理机构 广州知友专利商标代理有限公司 44104
代理人 宣国华

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

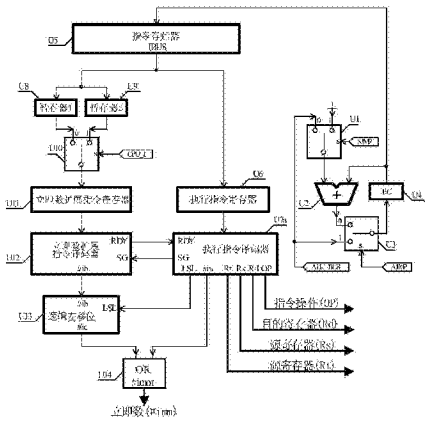
权利要求书2页 说明书8页 附图4页

(54)发明名称

一种将计算机指令中立即数扩展的方法和装置

(57)摘要

本发明公开了一种将计算机指令中立即数扩展的方法和装置，在RISC指令系统中至少新增一条立即数扩展指令，在编程时，将一个位数大于执行指令的立即数域的长度的立即数拆分成高位和低位立即数字段，将高位立即数字段经过有符号或无符号扩展后装入该立即数扩展指令的立即数域，而低位立即数字段则要刚好装满执行指令的立即数域；在取指令时，将该新增的立即数扩展指令和紧随其后的执行指令同步分别送入各自的译码器进行译码；在译码时，立即数扩展指令译码器输出的立即数经过逻辑左移位后再和执行指令译码器输出的立即数进行合并，合并结果得到一个位数大于执行指令的立即数域的长度的立即数。使用本发明的方法可以极大改善程序执行的效率，既节省时间又节省空间。



CN 105677298 A

1.一种将计算机指令中立即数扩展的方法,其特征是:

在现有的RISC计算机指令系统中至少增设一条立即数扩展指令和对应的一个立即数扩展指令译码器,该立即数扩展指令包含有操作码和立即数域;具体过程如下:

在指令编程过程中,将一个位数大于执行指令的立即数域的长度的立即数拆分成两个字段,一个是高位立即数字段,另一个是低位立即数字段;所述的高位立即数字段经过有符号数或无符号数扩展后再填入上述新增的立即数扩展指令的立即数域,所述的低位立即数字段则填入所述执行指令的立即数域,并且要求所述的低位立即数字段的长度必须等于所述执行指令的立即数域的长度;另外上述新增的立即数扩展指令要前置于携带该低位立即数字段的所述执行指令,中间不能插入别的指令;

在取指令过程中,取指令电路同步将上述新增的立即数扩展指令送入立即数扩展指令寄存器和将紧随该立即数扩展指令之后的执行指令送入执行指令寄存器;

在指令译码过程中,立即数扩展指令译码器对立即数扩展指令寄存器输出的指令进行译码的同时,执行指令译码器也对执行指令寄存器输出的指令进行译码,立即数扩展指令译码器输出的立即数经过逻辑左移位后再和执行指令译码器输出的立即数进行合并,合并结果得到一个位数大于执行指令的立即数域的长度的立即数;该立即数的数值与上述的指令编写过程中的被拆分的立即数的数值相等。

2.根据权利要求1的方法,其特征是:所述立即数扩展指令是一条非执行指令,只到译码这一阶段就终结,它不需要经历执行、回写的阶段。

3.根据权利要求1的方法,其特征是:通过在取指令电路中设置一个暂存器,将所述新增的立即数扩展指令延迟一条指令位置,以等待紧随其后的执行指令的到来,使得取指令电路将上述新增的立即数扩展指令和紧随其后的执行指令能够同步分别送入各自的译码器进行译码;所述的暂存器是寄存器。

4.根据权利要求1的方法,其特征是:所述立即数扩展指令译码器在译码时受控于执行指令译码器送来的一个符号数标志信号,最后译码输出一个立即数和一个译码状态标志信号,并且执行如下过程:如果上述新增的立即数扩展指令寄存器输出的指令是上述新增的立即数扩展指令,则设置该译码状态标志信号为真;否则,如果上述新增的立即数扩展指令寄存器输出的指令不是上述新增的立即数扩展指令,则设置该译码状态标志信号为假和设置该立即数扩展指令译码器输出的立即数为0值;所述的执行指令译码器送来的符号数标志信号是用于选择该立即数扩展指令译码器译码输出的立即数是有符号数还是无符号数。

5.根据权利要求4的方法,其特征是:所述执行指令译码器在译码过程中根据当前译码的执行指令的操作码来译码输出一个逻辑左移位的位数和一个符号数标志信号,所述的逻辑左移位的位数等于其当前译码的执行指令的立即数域的长度,所述的符号数标志信号就是当前译码的执行指令操作码所表示的立即数是有符号数还是无符号数;另外受控于上述的立即数扩展指令译码器输出的译码状态标志信号,其受控的过程是:如果该译码状态标志信号为真,则执行指令译码器译码输出的立即数是一个无符号数;否则如果该译码状态标志信号为假,则执行指令译码器将按照操作码的要求译码输出一个有符号或无符号立即数。

6.一种实现权利要求1所述方法的装置,包括取指令电路、执行指令寄存器、执行指令译码器,所述执行指令译码器,其用于对执行指令寄存器输出的指令进行译码,其特征是还

包括：

一个立即数扩展指令寄存器，用于存贮由取指令电路送来的指令；

一个立即数扩展指令译码器，用于对上述的立即数扩展指令寄存器输出的指令进行译码，另外其还受控于上述的改进的执行指令译码器输出的一个符号数标志信号，最后译码输出一个立即数和一个译码状态标志信号；

一个逻辑左移位电路，用于接收来自执行指令译码器输出的逻辑左移位的位数，并能按照该逻辑左移位的位数的要求完成对上述的立即数扩展指令译码器输出的立即数进行逻辑左移位，结果输出一个立即数；

一个数据合并电路，用于将上述逻辑左移位电路输出的立即数和改进的执行指令译码器输出的立即数进行逻辑或运算、或者是加法运算，运算结果输出一个立即数；

所述执行指令译码器还受控于上述的立即数扩展指令译码器输出的译码状态标志信号，最后译码输出一个立即数、一个逻辑左移位的位数和一个符号数标志信号。

7.根据权利要求6的装置，其特征在于所述取指令电路中还设有用于将立即数扩展指令延迟一条指令位置的暂存器。

一种将计算机指令中立即数扩展的方法和装置

技术领域

[0001] 本发明属于计算机领域，具体涉及一种将计算机指令中立即数扩展的方法和装置，使得RISC计算机指令可以获得位数大于执行指令立即数域的立即数。

背景技术

[0002] 当前，计算机系统主要分为CISC系统和RISC系统，对应于这两种系统分别有两种不同风格的指令规格。CISC系统出现最早，其指令集特点是规模比较大，指令能够携带的立即数可以很大，而且各指令长短不一致，最具代表性就是X86指令集；而RISC系统出现比较晚，其指令集特点是规模比较小，只包含有常用的指令，而且各指令长度都是一致，指令能够携带的立即数比较小，最具代表性就是ARM指令集和MISP指令集。虽然CISC系统和RISC系统的指令风格不同，但是它们共同点都是在指令中包含有操作码和若干操作数，这些操作数可能是寄存器名也可能是立即数。计算机指令结构如表1。其中操作码用于控制译码器，使译码器输出执行该指令的操作开关，同时也分离出各操作数。

[0003] 表1计算机指令结构

[0004]

操作码	若干操作数(或立即数)

[0005] 当前计算机系统常用图1所示的电路对指令进行读取和译码处理。

[0006] 在图1中，取指令电路是由U1、U2、U3、U4构成，其中U1是数据选择器，其用于选择相对跳转PC增量，U2是加法器，完成下一条指令地址的计算，U3是数据选择器，其用于选择下一条指令的指针，U4是PC寄存器，其存有指令地址值；U5是程序存贮器，里面存有程序指令；U6执行指令寄存器；U7是执行指令译码器，其功能是对执行指令寄存器输出的指令进行译码。

[0007] 在图1中，其电路的工作过程是：

[0008] 第一步是取指令阶段：由U1、U2、U3、U4完成下一条指令地址计算后PC(U4)向程序存贮器(U5)输出指令的地址值，然后程序存贮器(U5)在其输出总线上输出指令。

[0009] 第二步是译码阶段：程序存贮器(U5)输出的指令被锁存到执行指令寄存器(U6)中，执行指令译码器(U7)直接对执行指令寄存器(U6)里的指令进行译码，译码出该指令的操作信号(OP)、目的寄存器号(Rd)、源寄存器号(Rs)、源寄存器号(Rt)、立即数(#imm)等信息。

[0010] 从上面描述的指令读取和译码处理过程可以看出：在CISC指令系统中，由于CISC数据处理指令可以携带几个字节的立即数，所以对于CISC指令系统来说只要取指令电路动作足够快，获取大立即数根本不是难事，坏处是由于各条指令长度不一致(从1字节到15字节不等)，为了读取完整的指令，取指令电路必须分析每条指令的操作码才能确定该指令的长度，然后才能读取完整的指令，显然CISC的取指令电路相当复杂了，再者由于CISC数据处理指令需要表达携带几个字节立即数长度，所以CISC指令集数量相当庞大，比RISC指令规模大得多，好处是能够节省指令存贮器的空间。而RISC指令系统就不同了，由于在RISC的指

令系统中指令的长度规格只有1、2种，比如32位的ARM、MIPS，其指令长度都是32位(即4个字节)，其好处是可以简化取指令电路，使得取指令电路不用分析指令的操作码就可以读取完整的指令。但是由于指令长度不长，所以RISC的指令可以携带的立即数一般都不大，比如ARM的数据运算指令携带的立即数最大是12位，而MIPS指令系统的数据运算指令携带的立即数最大是16位，要这些RISC指令系统携带32位的立即数是不可能的，因为其指令长度只有32位，所以在RISC指令系统中要取得32位立即数就费不少周折了，目前常用的方法有两种：第一种是分段读取法，具体是先读取高16位的立即数，再读取低16位的立即数，从而组成32位立即数，为了支持这种分段读取法，几乎所有的RISC指令系统都是专门制定了相应的指令，比如ARM的MOVW和MOVT指令、MIPS的LUI和ADDIU指令；第二种是访存法，具体是把32位立即数当做常数放在离当前PC值不太远的地方，然后使用一条与PC相关的访存指令读取它。但是不管使用哪种读取方法，要获取一个32位的立即数都是费时，怎么都要花2个周期才能完成。

[0011] 比如要处理表达式：R2＝R3&0x12345678，其中立即数0x12345678是一个32位无符号数，对于MIPS来说，由于逻辑与运算指令ANDI$d,$s,#imm16的立即数域只有16位，而上述表达式立即数0x12345678的位数显然大于16位，32位立即数0x12345678是无法装入指令ANDI$d,$s,#imm16的立即数域，所以无法直接使用指令ANDI$d,$s,#imm16，只能使用指令AND$d,$s,$t。那么当前MIPS的做法是：

[0012] 1、使用分段读取法的是：

[0013] LUI $1,#1234H;(取得高16位常数，要花1个周期时间)

[0014] ADDIU $1,#5678H;(取得低16位常数，要花1个周期时间)

[0015] AND $2,$3,$1；

[0016] 2、使用访存法的是：

[0017] LWPC $1,[PC,#offest];(这条指令要花2个周期时间)

[0018] AND $2,$3,$1；

[0019] 由此可见，无论使用哪种方法，MIPS处理R2＝R3&0x12345678共需要花3个周期时间，其中获取32位常数0x12345678需要花2个周期时间。

发明内容

[0020] 为了解决现有的RISC指令系统获取大立即数执行效率低下的不足，又要维持RISC指令系统精简、统一长度的优点，本发明提出了一种将计算机指令中立即数扩展的方法，利用这种方法RISC计算机指令也能够快速获得一个位数超过执行指令立即数域的立即数。

[0021] 一种将计算机指令中立即数扩展的方法，在现有的RISC计算机指令系统中至少增设一条立即数扩展指令和对应的一个立即数扩展指令译码器，该立即数扩展指令包含有操作码和立即数域；具体过程如下：

[0022] 在指令编写过程中，将一个位数大于执行指令的立即数域的长度的立即数分成两个字段，一个是高位立即数字段，另一个是低位立即数字段；所述的高位立即数字段经过有符号数或无符号数扩展后再填入上述新增的立即数扩展指令的立即数域，所述的低位立即数字段则填入所述执行指令的立即数域，并且要求所述的低位立即数字段的长度等于所述执行指令的立即数域的长度；另外上述新增的立即数扩展指令要前置于携带该低位立即数

字段的所述执行指令，中间不能插入别的指令；

[0023] 在取指令过程中，取指令电路同步将上述新增的立即数扩展指令送入立即数扩展指令寄存器和将紧随该立即数扩展指令之后的执行指令送入执行指令寄存器；

[0024] 在指令译码过程中，立即数扩展指令译码器对立即数扩展指令寄存器输出的指令进行译码的同时，执行指令译码器也对执行指令寄存器输出的指令进行译码，立即数扩展指令译码器输出的立即数经过逻辑左移位后再和执行指令译码器输出的立即数进行合并，合并结果得到一个位数大于执行指令的立即数域的长度的立即数，该立即数的数值与上述的指令编写过程中的被拆分的立即数的数值相等。

[0025] 所述立即数扩展指令是一条非执行指令，只到译码这一阶段就终结，它不需要经历执行、回写的阶段。

[0026] 通过在取指令电路中设置一个暂存器，将所述新增的立即数扩展指令延迟一条指令位置，以等待紧随其后的执行指令的到来，使得取指令电路将上述新增的立即数扩展指令和紧随其后的执行指令能够同步分别送入各自的译码器进行译码；所述的暂存器是寄存器。

[0027] 所述立即数扩展指令译码器在译码时受控于执行指令译码器送来的一个符号数标志信号，最后译码输出一个立即数和一个译码状态标志信号，并且执行如下过程：如果上述新增的立即数扩展指令寄存器输出的指令是上述新增的立即数扩展指令，则设置该译码状态标志信号为真；否则，如果上述新增的立即数扩展指令寄存器输出的指令不是上述新增的立即数扩展指令，则设置该译码状态标志信号为假和设置该立即数扩展指令译码器输出的立即数为0值；所述的执行指令译码器送来的符号数标志信号是用于选择该立即数扩展指令译码器译码输出的立即数是有符号数还是无符号数。

[0028] 所述执行指令译码器在译码过程中根据当前译码的执行指令的操作码来译码输出一个逻辑左移位的位数和一个符号数标志信号，所述的逻辑左移位的位数等于其当前译码的执行指令的立即数域的长度，所述的符号数标志信号是表示当前译码的执行指令操作码所表示的立即数是有符号数还是无符号数；另外受控于上述的立即数扩展指令译码器输出的译码状态标志信号，其受控的过程是：如果该译码状态标志信号为真，则执行指令译码器译码输出的立即数是一个无符号数；否则如果该译码状态标志信号为假，则执行指令译码器将按照操作码的要求译码输出一个有符号或无符号立即数。

[0029] 一种实现所述方法的装置，包括取指令电路、执行指令寄存器、执行指令译码器，所述执行指令译码器，其用于对执行指令寄存器输出的指令进行译码，其特征是：

[0030] 一个立即数扩展指令寄存器，用于存贮由取指令电路送来的指令；

[0031] 一个立即数扩展指令译码器，用于对上述的立即数扩展指令寄存器输出的指令进行译码，另外其还受控于上述的改进的执行指令译码器输出的一个符号数标志信号，最后译码输出一个立即数和一个译码状态标志信号；

[0032] 一个逻辑左移位电路，用于接收来自执行指令译码器输出的逻辑左移位的位数，并能按照该逻辑左移位的位数的要求完成对上述的立即数扩展指令译码器输出的立即数进行逻辑左移位，结果输出一个立即数；

[0033] 一个数据合并电路用于将上述逻辑左移位电路输出的立即数和改进的执行指令译码器输出的立即数进行逻辑或运算、或者加法运算，运算结果输出一个立即数；

[0034] 所述执行指令译码器还受控于上述的立即数扩展指令译码器输出的译码状态标志信号，最后译码输出一个立即数、一个逻辑左移位的位数和一个符号数标志信号。

[0035] 所述取指令电路中还设有用于将上述的立即数扩展指令延迟一条指令位置的暂存器。

[0036] 通过使用本发明的方法，同样要处理背景技术所述的表达式：R2＝R3&0x12345678，其中立即数0x12345678是一个32位无符号数。对于MIPS来说，这里可以直接使用MPS的执行指令ANDI$d,$s,#imm16，其方法是：将32位无符号立即数0x12345678拆分为高位立即数字段1234h和低位立即数字段5678h，其中低位立即数字段5678h是16位。将高位立即数字段1234h经无符号数扩展后填入上述新增的立即数扩展指令的立即数域，而低位立即数字段5678h刚好填满执行指令ANDI$d,$s,#imm16的立即数域，结果如下：

[0037] HIMM #1234h;(这是新增的立即数扩展指令，其携带高位立即数字段#1234h)

[0038] ANDI $2,$3,#5678h;(执行指令携带低16位立即数字段#5678h)

[0039] 其执行的效果等同于：ANDI$2,$3,#12345678h;

[0040] 如果程序存贮器每次只能在其接口总线上输出1条指令----即单倍速取指令模式，则处理R2＝R3&0x12345678共需要花2个周期时间，其中立即数扩展指令(HIMM#1234h)需要延迟1条指令的时间，以等侯紧随其后的执行指令(ANDI$2,$3,#5678h)的到来，才能同步译码。

[0041] 如果程序存贮器每次都能在其接口总线上输出2条指令----即双倍速取指令模式，则处理R2＝R3&0x12345678仅需要花1个周期时间，此时立即数扩展指令(HIMM#1234h)不需要任何延迟就可以和紧随其后的执行指令(ANDI$2,$3,#5678h)同步译码，所以它不花任何时间！

[0042] 从上面例子可以看出，使用本发明的方法可以极大改善程序执行的效率，既节省时间又节省空间。

附图说明

[0043] 图1为现有计算机系统的取指令和指令译码电路示意图。

[0044] 图2为举例说明本发明方法的大立即数拆分与合并的示意图。

[0045] 图3为程序存贮器输出接口总线宽度等于1条指令宽度的情形的实施例电路示意图。

[0046] 图4为程序存贮器输出接口总线宽度等于2条指令宽度的情形的实施例电路示意图。

具体实施方式

[0047] 本发明方法主要是通过在在现有的RISC计算机指令系统中增设一条立即数扩展指令和对应的一个立即数扩展指令译码器，该立即数扩展指令包含有操作码和立即数域；立即数扩展指令是一条非执行指令，只到译码这一阶段就终结，它不需要经历执行、回写的阶段。在指令编写过程中，将一个位数大于执行指令的立即数域的长度的立即数分成两个字段，一个是高位立即数字段，另一个是低位立即数字段；高位立即数字段经过有符号数或无符号数扩展后再填入上述新增的立即数扩展指令的立即数域，低位立即数字段则填入所

述执行指令的立即数域，并且要求低位立即数字段的长度等于所述执行指令的立即数域的长度；另外上述新增的立即数扩展指令要前置于携带该低位立即数的所述执行指令，中间不能插入别的指令；在图2中以一个32位有符号立即数0x12345678为例子加以说明一个大的立即数拆分与合并的过程。在编程时，将一个位数大于执行指令的立即数域的长度的立即数分成两个字段，一个是高位立即数字段，另一个是低位立即数字段，低位立即数字段的长度等于该执行指令的立即数域的长度，剩下的高位立即数字段要经过有符号数或无符号数扩展后再填入上述新增的立即数扩展指令的立即数域。在译码时，上述新增的立即数扩展指令译码器输出的立即数需要逻辑左移位的位数等于该执行指令的立即数域的长度，也就是说，如果执行指令的立即数域为n位，则拆分得到的低位立即数字段是n位，上述新增的立即数扩展指令译码器输出的立即数需要逻辑左移位的位数等于n位。在实际应用中，当所需的立即数的长度大于执行指令的立即数域时才会使用到上述新增的立即数扩展指令，所以上述改进的执行指令译码器在译码时必须要检查是否有上述新增的立即数扩展指令在上述新增的立即数扩展指令译码器同步译码，如果上述改进的执行指令译码器在译码时发现有上述新增的立即数扩展指令在上述新增的立即数扩展指令译码器同步译码，说明有高位立即数字段，那么上述改进的执行指令译码器译码输出的立即数是无符号数，否则如果上述改进的执行指令译码器在译码时没发现有上述新增的立即数扩展指令在上述新增的立即数扩展指令译码器同步译码，说明没有高位立即数字段，那么上述改进的执行指令译码器将按照操作码的要求译码输出的立即数是有符号或无符号立即数。

[0048] 在图2中，由于执行指令5的立即数域是16位和新增的立即数扩展指令4的立即数域为26位，那么将32位有符号立即数1(0x12345678)拆分为一个26位有符号数的高位立即数字段2(0x1234)和一个16位无符号的低位立即数字段3(0x5678)，高位立即数字段2(0x1234)嵌入新增的立即数扩展指令4的立即数域(Immediate)得到立即数扩展指令6，而低位立即数字段3(0x5678)嵌入执行指令5的立即数域(Immediate)得到执行指令7。取指令时，上述改进的取指令电路同步地把新增的立即数扩展指令6送入新增的立即数扩展指令寄存器(U11)和把执行指令7送入执行指令寄存器(U6)。新增的立即数扩展指令译码器(U12)对U11的指令进行译码，译码得到32位有符号立即数8(0x1234)，而改进的执行指令译码器(U7a)对U6的指令进行译码，译码得到32位无符号立即数10(0x5678)。立即数8经过新增的逻辑左移位电路(U13)进行逻辑左移位16位后得到32位的有符号立即数9(0x12340000)，新增的数据合并电路(U14)将立即数9和立即数10合并(即逻辑或运算)后得到32位有符号立即数11(0x12345678)。立即数11和立即数1完全相同。需要说明的是：如果要拆分的立即数1是无符号数，则高位立即数字段要进行无符号数扩展后再装入新增的立即数扩展指令4的立即数域；否则，如果要拆分的立即数1是有符号数，则高位立即数字段要进行有符号数扩展后再装入新增的立即数扩展指令4的立即数域。

[0049] 在现实应用中，由于程序存贮器输出接口总线宽度有等于1条指令的宽度，也有等于2条指令的宽度，或甚至等于更多条指令的宽度，所以本实施例子分为两个情形说明：第一种应用情形是程序存贮器输出接口总线宽度等于1条指令的宽度；第二种应用情形是程序存贮器输出接口总线宽度等于2条指令的宽度。对于程序存贮器输出接口总线宽度大于2条指令宽度的情形可以参考第二种应用情形的方法改进现有的取指令电路，在此不再说明。

[0050] 对于上述的第一种应用情形，其实施电路如图3所示。

[0051] 在图3中，取指令电路是由U1、U2、U3、U4、U8、U9、U10等电路构成，其中U1、U2、U3、U4与图1的作用相同，此处不再重复描述。U5是程序存贮器。U8、U9、U10、U11、U12、U13、U14是在图1的基础上新增的电路，其中U8、U9都是暂存器，这2个暂存器都是寄存器，其作用是让流过它们的指令能够延迟一条指令位置，其中一个暂存器工作于程序正常执行态，而另一个暂存器则是工作于取指令页故障异常执行态。U10是数据选择器，系统根据当前程序处于正常执行态还是处于取指令页故障异常执行态来控制U10，使得U10选择相应的暂存器作为当前工作的暂存器，不是当前工作的暂存器就处于停止状态，如果U8处于工作状态，则U10选中U8作为其数据输入，反之，如果U9处于工作状态，则U10选中U9作为其数据输入，这样做可以实现当出现取指令页故障时系统能够切换到工作于取指令页故障异常执行态的暂存器，从而保护已经被送入工作于程序正常执行态的暂存器的上述新增的立即数扩展指令。U6是现有的执行指令寄存器，用于存贮当前译码的指令。U7a是改进的执行指令译码器。U11是新增的立即数扩展指令寄存器。U12是新增的立即数扩展指令译码器。U13是新增的逻辑左移位电路。U14是新增的数据合并电路。

[0052] 在图3中，其电路的工作过程是：

[0053] 第一步是取指令阶段：由U1、U2、U3、U4完成下一条指令地址计算后PC(U4)向程序存贮器(U5)输出指令的地址值，然后程序存贮器(U5)在其输出总线上输出1条指令，这条指令送入执行指令寄存器(U6)，同时：如果此时U8处于工作状态，则U10选中U8作为其数据输入，U5接口总线上的指令送入U8，而U8旧的指令通过U10送入立即数扩展指令寄存器(U11)；如果此时U9处于工作状态，则U10选中U9作为其数据输入，U5接口总线上的指令送入U9，而U9旧的指令通过U10送入立即数扩展指令寄存器(U11)。

[0054] 第二步是译码阶段：立即数扩展指令译码器(U12)对立即数扩展指令寄存器(U11)输出的指令进行译码，并结合改进的执行指令译码器(U7a)输出的符号数标志信号(SG)的状态，译码输出一个立即数(#ib)和一个译码状态标志信号RDY，该译码状态标志信号RDY是这样定义的：如果U11输出的指令是上述新增的立即数扩展指令，则该译码状态标志信号RDY为真，U12译码输出的立即数(#ib)为有效的立即数；而如果U11输出的指令不是上述新增的立即数扩展指令，则该译码状态标志信号RDY为假，U12译码输出的立即数为0值。其中U7a输出的符号数标志信号(SG)是用于控制U12译码输出的立即数(#ib)是有符号数还是无符号数。上述新增的立即数扩展指令的任务到这里就完成了。改进的执行指令译码器(U7a)对执行指令寄存器(U6)输出的指令进行译码，它还结合立即数扩展指令译码器(U12)输出的译码状态标志信号RDY，译码输出该执行指令的执行操作信号(OP)、目的寄存器(Rd)、源寄存器(Rs)、源寄存器(Rt)、立即数(#ia)、符号数标志信号(SG)和逻辑左移位的位数(LSL)。如果立即数扩展指令译码器(U12)输出的译码状态标志信号RDY为真，则执行指令译码器(U7a)译码输出的立即数(#ia)是一个无符号数；否则立即数扩展指令译码器(U12)输出的译码状态标志信号RDY为假，则执行指令译码器(U12)将按照执行指令操作码的正常要求译码输出的立即数(#ia)。如果U6输出的指令是上述新增的立即数扩展指令，则U7a将按照空操作指令(即NOP指令)译码输出。U7a通过向逻辑左移位电路(U13)输出逻辑左移位的位数(LSL)而控制U13，使得U13按照该逻辑左移位的位数(LSL)的要求对U12输出的立即数(#ib)进行逻辑左移位，移位后的结果由U13输出立即数(#ic)，接着由数据合并电路(U14)

将U7a输出的立即数(#ia)和U13输出立即数(#ic)进行合并(即逻辑或运算或者是加法运算),最后得到立即数(#imm)。

[0055] 对于上述的第二种应用情形,其实施电路如图4所示。

[0056] 在图4中,取指令电路是由U1、U2、U3、U4、U8、U9、U10、U15、U16、U17、U18、U19等电路构成,其中U1、U2、U3、U4与图1的作用相同,只是U1输入的顺序执行的PC增量由常量(1)改为变量(APC)。U5是程序存贮器。U8、U9、U10、U11、U12、U13、U14、U15、U16、U17、U18、U19是在图1的基础上增加的电路,其中U8、U9都是暂存器,这2个暂存器都是寄存器,其作用是让流过它们的指令能够延迟一条指令位置,其中一个暂存器工作于程序正常执行态,而另一个暂存器则是工作于取指令页故障异常执行态。U10是数据选择器,系统根据当前程序处于正常执行态还是处于取指令页故障异常执行态来控制U10,使得U10选择相应的暂存器作为当前工作的暂存器,不是当前工作的暂存器就处于停止状态,如果U8处于工作状态,则U10选中U8作为其数据输入,反之,如果U9处于工作状态,则U10选中U9作为其数据输入,这样做可以实现当出现取指令页故障时系统能够切换到工作于取指令页故障异常执行态的暂存器,从而保护已经被送入工作于程序正常执行态的暂存器的上述新增的立即数扩展指令。U6是现有的执行指令寄存器,用于存贮当前译码的指令。U7a是改进的执行指令译码器。U11是新增的立即数扩展指令寄存器。U12是新增的立即数扩展指令译码器。U13是新增的逻辑左移位电路。U14是新增的数据合并电路。U15、U16是新增的数据选择器。U17、U18是新增的数值比较器。U19是新增的组合逻辑电路。

[0057] 在图4中,其电路的工作过程是:

[0058] 第一步是取指令阶段:由U1、U2、U3、U4完成下一条指令地址计算后PC(U4)向程序存贮器(U5)输出指令的地址值,然后程序存贮器(U5)在其接口总线IBUS0和IBUS1上分别同时输出2条指令,其中总线IBUS0输出偶数地址的指令,而总线IBUS1输出奇数地址的指令。为了表述方便,在此将总线IBUS0输出的指令称为第一条指令,而将总线IBUS1输出的指令称为第二条指令,从程序指令的位置看,第一条指令是位于第二条指令之前。第一条指令送到U15的端口0处,第一条指令的操作码送到数值比较器U18的一个输入端口,与U18另一输入端的上述新增的立即数扩展指令的操作码比较,如果比较结果为相等,则U18输出端A为1,否则A为0。第二条指令送到U16的端口1处,第二条指令的操作码送到数值比较器U17的一个输入端口,与U17另一输入端的上述新增的立即数扩展指令的操作码比较,如果比较结果为相等,则U17输出端B为1,否则B为0。如果此时U8处于工作状态,则U10选中U8作为其数据输入,第二条指令送入U8,U8旧的指令通过U10送入数据选择器U15的端口1处;如果此时U9处于工作状态,则U10选中U9作为其数据输入,第二条指令送入U9,U9旧的指令通过U10送入数据选择器U15的端口1处。指令再经过U15选择后送入立即数扩展指令寄存器(U11)。组合逻辑电路(U19)根据U18的输出信号A、U17的输出信号B以及程序指针寄存器PC输出的取指令地址的最低位PC.2的状态来实现如表2所示的逻辑关系并将结果输出到端口S1、S2、APC。其中信号PC.2为0时,表示PC指向第一条指令,而PC.2为1时,表示PC指向第二条指令。U19输出端口信号S1控制数据选择器(U15)完成数据选择后将指令输出到立即数扩展指令寄存器(U11),其中:如果S1为1,则选中U10输出的旧的指令为数据输入,否则选中第一条指令为数据输入。U19输出端口信号S2控制数据选择器(U16)完成数据输入选择后将指令输出到执行指令寄存器(U6),其中:如果S2为1,则选中第二条指令为数据输入,否则选中第一条指令为

数据输入。U19端口信号APC是顺序执行PC增量的数值：如果APC为1，则PC增量为1条指令，如果APC为2，则PC增量为2条指令。

[0059] 表2 U19组合逻辑电路真值表

[0060]

输入端			输出端		
PC.2	A	B	S1	S2	APC
0	0	0	1	0	1
0	0	1	1	0	2
0	1	x	0	1	2

[0061]

1	x	x	1	1	1

[0062] 第二步是译码阶段(这一阶段和图3的译码阶段相同)：立即数扩展指令译码器(U12)对立即数扩展指令寄存器(U11)输出的指令进行译码，并结合改进的执行指令译码器(U7a)输出的符号数标志信号(SG)的状态，译码输出一个立即数(#ib)和一个状态标志信号RDY，该状态标志信号RDY是这样定义的：如果U11输出的指令是上述新增的立即数扩展指令，则该状态标志信号RDY为真，U12译码输出的立即数(#ib)为有效的立即数；而如果U11输出的指令不是上述新增的立即数扩展指令，则该状态标志信号RDY为假，U12译码输出的立即数为0值。其中U7a输出的符号数标志信号(SG)是用于选择U12译码输出的立即数(#ib)是有符号数还是无符号数。上述新增的立即数扩展指令的任务到这里就完成了。改进的执行指令译码器(U7a)对执行指令寄存器(U6)输出的指令进行译码，它还结合立即数扩展指令译码器(U12)输出的状态标志信号RDY，译码输出该执行指令的执行操作信号(OP)、目的寄存器(Rd)、源寄存器(Rs)、源寄存器(Rt)、立即数(#ia)、符号数标志信号(SG)和逻辑左移位的位数(LSL)。如果立即数扩展指令译码器(U12)输出的译码状态标志信号RDY为真，则执行指令译码器(U7a)译码输出的立即数(#ia)是一个无符号数；否则立即数扩展指令译码器(U12)输出的译码状态标志信号RDY为假，则执行指令译码器(U12)将按照执行指令操作码的正常要求译码输出的立即数(#ia)。如果U6输出的指令是上述新增的立即数扩展指令，则U7a将按照空操作指令(即NOP指令)译码输出。U7a通过向逻辑左移位电路(U13)输出逻辑左移位的位数(LSL)而控制U13，使得U13按照该逻辑左移位的位数(LSL)的要求对U12输出的立即数(#ib)进行逻辑左移位，移位后的结果由U13输出立即数(#ic)，接着由数据合并电路(U14)将U7a输出的立即数(#ia)和U13输出立即数(#ic)进行合并(即逻辑或运算或者是加法运算)，最后得到立即数(#imm)。

[0063] 本发明的实施方式不限于此，按照本发明的上述内容，利用本领域的普通技术知识和惯用手段，在不脱离本发明的上述基本技术思想前提下，本发明还可以做出其它多种形式的修改、替换或变更，均落在本发明权利保护范围之内。

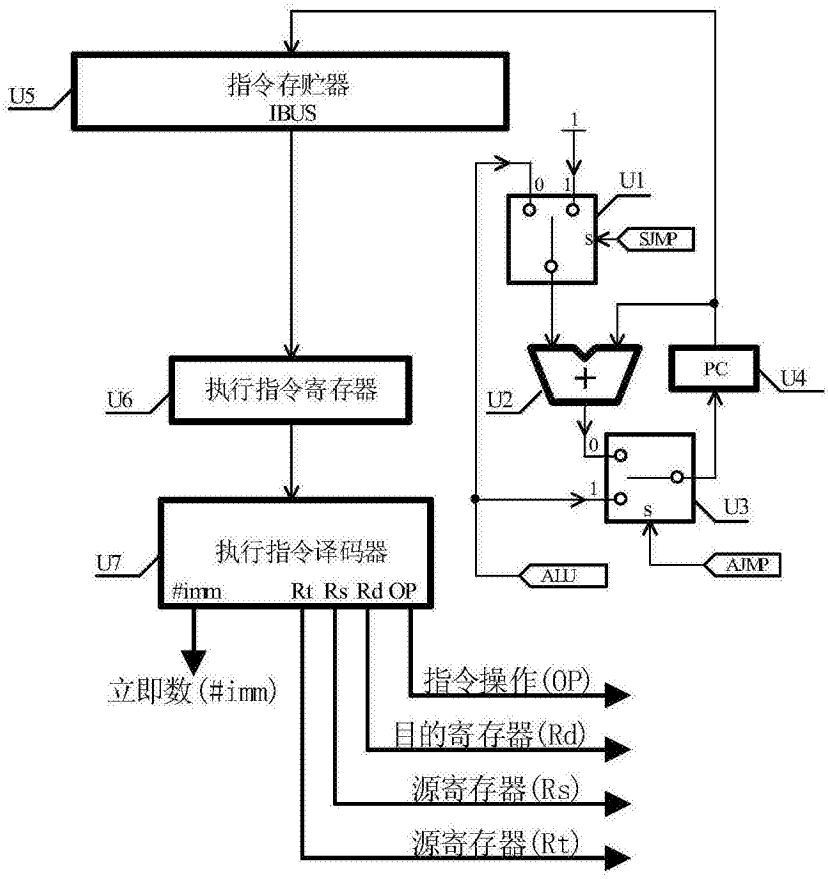

U5
指令存贮器
IBUS
1
0
1
U1
S
SJMP
U6
执行指令寄存器
U2
+
PC
U4
0
1
S
U3
U7
执行指令译码器
#imm
Rt Rs Rd OP
ALU
AJMP
立即数(#imm)
指令操作(OP)
目的寄存器(Rd)
源寄存器(Rs)
源寄存器(Rt)

图1

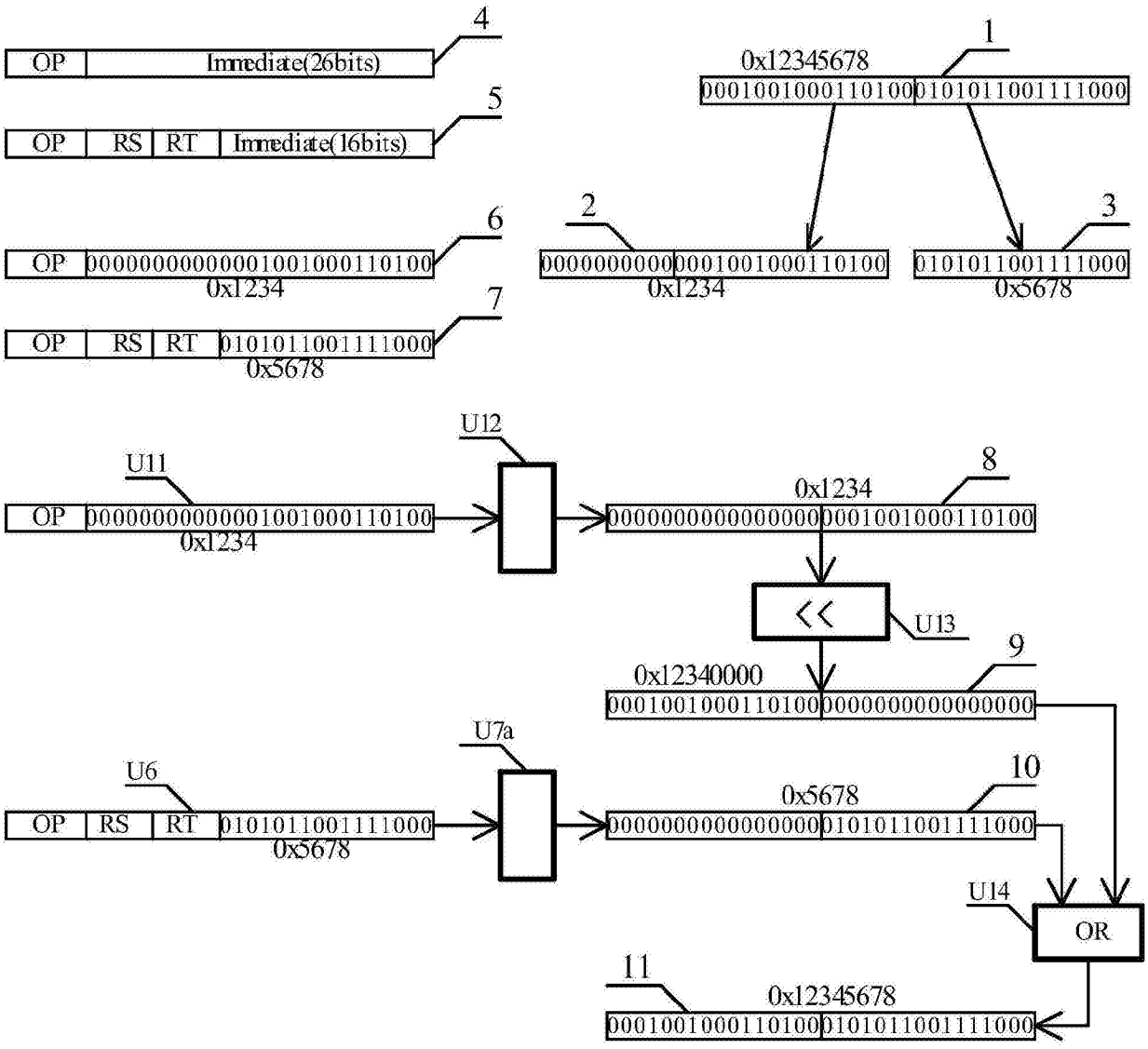

4
OP
Immediate(26bits)
5
OP
RS
RT
Immediate(16bits)
6
OP
00000000000001001000110100
0x1234
7
OP
RS
RT
0101011001111000
0x5678
1
0x12345678
00010010001101000101011001111000
2
00000000000001001000110100
0x1234
3
0101011001111000
0x5678
U12
U11
OP
00000000000001001000110100
0x1234
8
0x1234
00000000000000000001001000110100
<<
U13
9
0x12340000
00010010001101000000000000000000
U7a
U6
OP
RS
RT
0101011001111000
0x5678
10
0x5678
00000000000000000101011001111000
U14
OR
11
0x12345678
00010010001101000101011001111000

图2

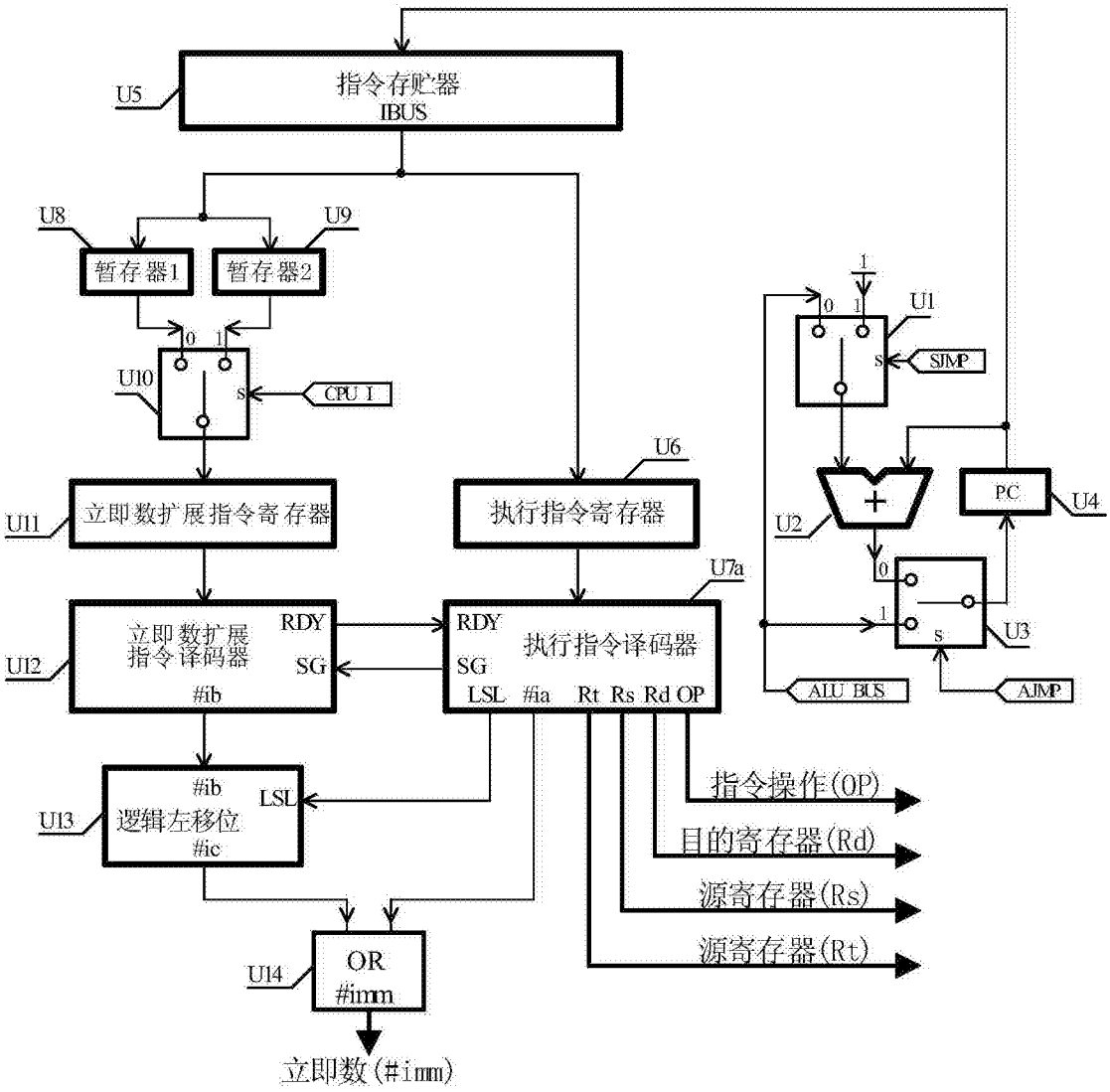

U5
指令存贮器
IBUS
U8
U9
暂存器1
暂存器2
0
1
U10
s
CPU I
U11
立即数扩展指令寄存器
U6
执行指令寄存器
U7a
U12
立即数扩展
指令译码器
#ib
RDY
SG
RDY
SG
执行指令译码器
LSL #ia Rt Rs Rd OP
U13
#ib
逻辑左移位
#ic
LSL
U14
OR
#imm
立即数(#imm)
指令操作(OP)
目的寄存器(Rd)
源寄存器(Rs)
源寄存器(Rt)
1
U1
SJMP
U2
PC
U4
U3
ALU BUS
AJMP

图3

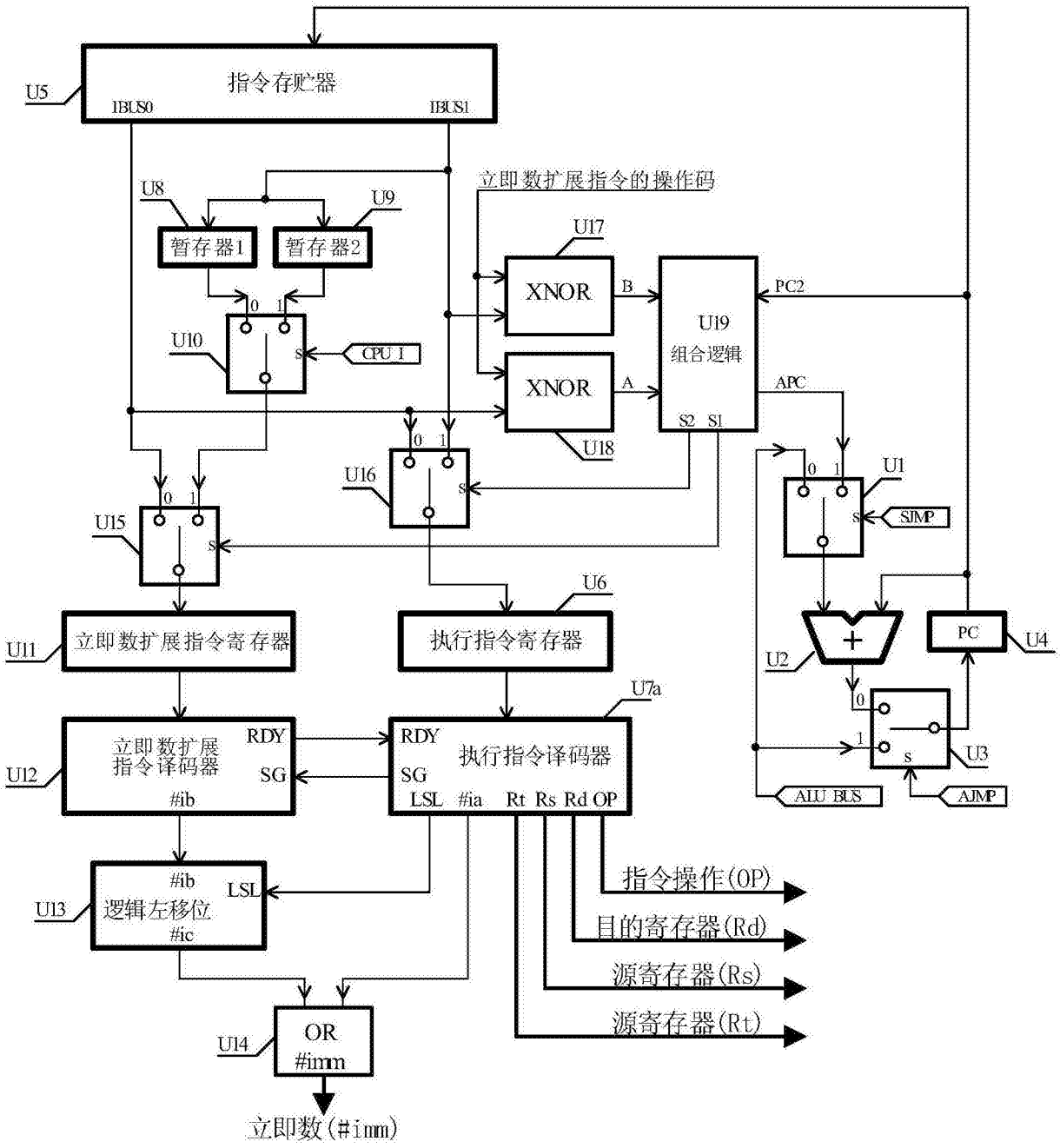

U5
指令存贮器
IBUS0
IBUS1
立即数扩展指令的操作码
U8
U9
暂存器1
暂存器2
U17
XNOR
B
U19
组合逻辑
PC2
U10
CPU 1
XNOR
A
APC
S2 S1
U18
U16
U1
SIMP
U15
U6
U11
立即数扩展指令寄存器
执行指令寄存器
U2
PC
U4
U7a
U12
立即数扩展
指令译码器
#ib
RDY
SG
执行指令译码器
LSL #ia Rt Rs Rd OP
U3
AJMP
ALU BUS
U13
逻辑左移位
#ic
LSL
指令操作(OP)
目的寄存器(Rd)
源寄存器(Rs)
源寄存器(Rt)
U14
OR
#imm
立即数(#imm)

图4