(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104915182 B
(45)授权公告日 2017.12.01

(21)申请号 201510233155.4

(22)申请日 2011.09.29

(65)同一申请的已公布的文献号
申请公布号 CN 104915182 A

(43)申请公布日 2015.09.16

(30)优先权数据
1019798.6 2010.11.23 GB

(62)分案原申请数据
201180056035.1 2011.09.29

(73)专利权人 ARM 有限公司
地址 英国剑桥

(72)发明人 戴维·詹姆斯·西尔
理查德·罗伊·格里森思怀特
奈杰尔·约翰·斯蒂芬斯

(74)专利代理机构 北京东方亿思知识产权代理有限责任公司 11258
代理人 李晓冬

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

(56)对比文件
CN 1799025 A,2006.07.05,
CN 1898873 A,2007.01.17,
US 6625724 B1,2003.09.23,
US 7885992 B2,2011.02.08,
US 7370180 B2,2008.05.06,
US 5922067 A,1999.07.13,
US 6678806 B1,2004.01.13,

审查员 张力

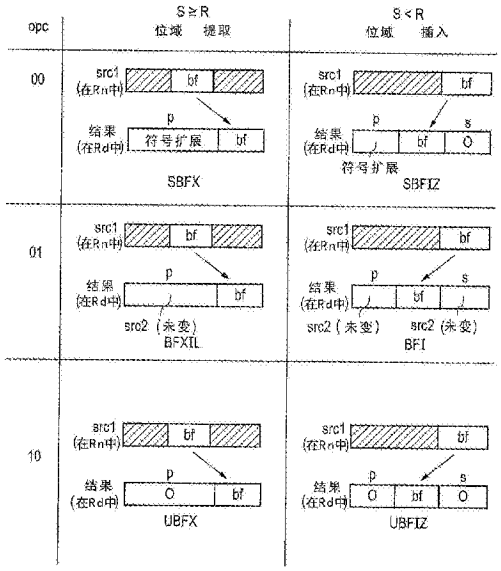
权利要求书2页 说明书28页 附图12页

(54)发明名称

具有位域操纵指令的数据处理装置及方法

(57)摘要

本申请涉及一种具有位域操纵指令的数据处理装置及方法，该数据处理装置(2)包含一处理电路(4)及指令译码器(6)。位域操纵指令控制该处理装置(2)从相应第一源数据元素src1及第二源数据元素src2产生至少一个结果数据元素。每一个结果数据元素包括对应于该相应第一源数据元素src1的位域bf的部分。比所插入的位域bf更有效的结果数据元素的位具有前缀值p，基于由该指令所指定的控制值选择该前缀值p，作为第一前缀值、第二前缀值及第三前缀值中的一个前缀值，该第一前缀值具有零值，该第二前缀值具有该相应第二源数据元素src2的一部分的值，该第三前缀值对应于该第一源数据元素src1的该位域bf的符号扩展。



CN 104915182 B

1.一种数据处理装置,包含:

处理电路,该处理电路被配置为执行处理操作;

指令译码器,该指令译码器响应于程序指令产生用于控制该处理电路执行所述处理操作的控制信号;其中:

所述程序指令包括至少一个指令,该至少一个指令指定一控制值,该控制值具有第一部分及第二部分,该第一部分用于指示选自复数个数据大小的选定数据大小,该第二部分用于指示至少一个控制参数,该至少一个控制参数具有根据该选定数据大小而变化的位数目,该第一部分及该第二部分各自具有可变数目的位;以及

该指令译码器响应于该至少一个指令产生用于控制该处理电路根据该选定数据大小及该至少一个控制参数执行相应处理操作的控制信号;

其中当处理该至少一个指令时,该指令译码器及该处理电路中的至少一者被配置为识别由该控制值的该第一部分包含的位数目,且根据由该第一部分包含的该位数目来识别:(a)该选定数据大小,以及(b)该控制值中的哪些位形成用于指示该至少一个控制参数的该第二部分。

2.如权利要求1所述的数据处理装置,其中随着该第二部分的位数目根据该选定数据大小而减小,该第一部分的位数目根据该选定数据大小而增大。

3.如权利要求1和2中任一项所述的数据处理装置,其中对于所述复数个数据大小的至少一子集,该第一部分至少包含具有第一状态的第一位及具有第二状态的X个剩余位,其中X为大于0或等于0的可变整数;并且

该指令译码器及该处理电路中的该至少一者被配置为:根据该控制值的预定部分内该第一位的位位置,识别由该控制值的该第一部分包含的位数目。

4.如权利要求3所述的数据处理装置,其中该第一部分包含至少一个额外位,该至少一个额外位用于提供用于识别该选定数据大小的其他信息。

5.如权利要求1至2中任一项所述的数据处理装置,其中该至少一个控制参数包含复数个控制参数;以及

当处理该至少一个指令时,该指令译码器及该处理电路中的该至少一者被配置为:根据由该第一部分包含的位数目来识别该第二部分中的哪些位指示所述复数个控制参数中的每一个。

6.如权利要求5所述的数据处理装置,其中所述复数个控制参数至少包含第一控制参数及第二控制参数,该第一控制参数具有随该选定数据大小增大而增大的位数目,该第二控制参数具有随该选定数据大小增大而减小的位数目;以及

该第二部分包括用于指示该第一控制参数及该第二控制参数的子部分,该指令译码器及该处理电路中的该至少一者根据由该第一部分包含的位数目来识别:该子部分中的哪些位指示该第一控制参数及该子部分中的哪些位指示该第二控制参数。

7.如权利要求1至2中任一项所述的数据处理装置,其中该至少一个指令包括至少一个单个指令多个数据(SIMD)指令,该至少一个SIMD指令识别包含至少一个源数据元素的源数据值;以及

对于该至少一个单个指令多个数据(SIMD)指令,该选定数据大小指示该至少一个源数据元素的数据元素大小,且该相应处理操作包含:对该源数据值的该至少一个源数据元素

中的每一个执行选定处理操作。

8.如权利要求7所述的数据处理装置，其中该至少一个单个指令多个数据(SIMD)指令包括位域操纵指令，且对于该位域操纵指令：

该相应处理操作包含：产生结果数据值，该结果数据值包含至少一个结果数据元素，每一个结果数据元素对应于该源数据值中的相应源数据元素；

每一个结果数据元素包含结果位域，该结果位域具有与该相应源数据元素内连续位的源位域的位值相对应的位值；以及

该至少一个控制参数指示由该源位域及该结果位域包含的位数目、该相应源数据元素内的该源位域的位置及该结果数据元素内的该结果位域的位置。

9.如权利要求8所述的数据处理装置，其中对于该位域操纵指令，若该源数据值包含复数个源数据元素且该结果数据值包含复数个结果数据元素，则该至少一个控制参数亦包括数据元素排序参数，该数据元素排序参数用于指示用于将所述复数个结果数据元素布置于该结果数据值内的次序。

10.如权利要求1至2中任一项所述的数据处理装置，其中该至少一个指令包括按位逻辑指令，该按位逻辑指令识别源数据值；且对于该按位逻辑指令：

该相应处理操作包含：产生结果数据值，该结果数据值等效于将逻辑运算应用于该源数据值及基于该控制值确定的第二数据值的结果；

该选定数据大小指示由该第二数据值内的重复模式的位包含的位数目；以及

该至少一个控制参数指示用于识别所述重复模式的位的位值的信息。

11.如权利要求10所述的数据处理装置，其中该按位逻辑运算包含与运算、或运算及异或运算中的一者。

12.一种用于处理装置响应于程序指令执行处理操作的数据处理方法，包含：

接收至少一个指令，该至少一个指令指定一控制值，该控制值具有第一部分及第二部分，该第一部分用于指示选自复数个数据大小的选定数据大小，该第二部分用于指示至少一个控制参数，该至少一个控制参数具有根据该选定数据大小而变化的位数目，该第一部分及该第二部分各自具有可变数目的位；

响应于该至少一个指令，识别由该控制值的该第一部分包含的位数目；

根据由该第一部分包含的位数目，识别：(a)该选定数据大小，以及(b)该控制值中的哪些位形成用于指示该至少一个控制参数的该第二部分；以及

控制该处理装置根据该选定数据大小及该至少一个控制参数执行相应处理操作。

具有位域操纵指令的数据处理装置及方法

[0001] 本申请是基于申请号为201180056035.1、申请日为2011年9月29日、申请人为ARM有限公司、发明名称为“具有位域操纵指令的数据处理装置及方法”的发明提出的分案申请。

技术领域

[0002] 本发明关于数据处理的领域。

背景技术

[0003] 在数据处理系统中，数据值有时可含有若干相邻位，这些位的值独立于数据值的其余位而受关注。这样一组相邻位可被认为是位域(bitfield)，且该组中的相邻位的数目可被认为是位域的宽度。例如，表示RGB色值的数据值可包括分别表示红色成分、绿色成分及蓝色成分的位域。有时，可能希望操纵含有位域的数据值以便将位域与该位域的周围隔离。例如，有人可能希望仅提取或替换RGB色值中的绿色成分。本技术试图提供位域操纵指令的有效编码，该位域操纵指令可控制处理装置执行各式各样不同种类的位域操纵。

[0004] 在本申请案中，记法<X:Y>指示从位位置X处的最高有效位延伸至位位置Y处的最低有效位的一组相邻位。由<X:Y>所描述的组宽度为X-Y+1位。例如，表述<4:0>指示位位置4至0处的5位宽度，其中位<4>比位<0>更有效。应注意，记法<X:Y>并非暗示储存位置内的位的任何特定储存次序，因为储存次序不必与有效性的次序相同(例如，可使用大端或小端储存格式)。

[0005] 在本申请案中，跟随有一系列1及0的前缀0b表示二进制记数法中的数值。例如，0b110表示十进制记数法中的值6。

发明内容

[0006] 从一个方面来看，本发明提供一种数据处理装置，该数据处理装置包含：

[0007] 处理电路，该处理电路被配置为执行处理操作；

[0008] 指令译码器，该指令译码器响应于程序指令产生用于控制该处理电路执行所述处理操作的控制信号；其中：

[0009] 所述程序指令包括位域操纵指令，该位域操纵指令至少识别第一源数据值、第二源数据值及控制值，该第第一源数据值包含各自具有N位<N-1:0>的至少一个第一源数据元素，该第二源数据值包含各自具有N位<N-1:0>的至少一个第二源数据元素；

[0010] 该控制值包括用于指示位域宽度W、源位位置A及结果位位置B的信息，其中1≤W≤N，0≤A≤(N-W)及0≤B≤(N-W)；以及

[0011] 该指令译码器响应于该位域操纵指令以产生用于控制该处理电路产生结果数据值的控制信号，该结果数据值包含至少一个结果数据元素，每个结果数据元素对应于相应第一源数据元素及相应第二源数据元素，每个结果数据元素具有N位<N-1:0>，这些位包含：

[0012] (a)具有与该相应第一源数据元素的位<A+W-1:A>相对应的位值的位<B+W-1:B>；

[0013] (b)若B+W＜N,则为具有与前缀值相对应的位值的位<N-1:B+W>,根据该控制值选择该前缀值为以下中的一者:(i)第一前缀值,该第一前缀值包含各自具有零值的位,(ii)第二前缀值,该第二前缀值具有该相应第二源数据元素的位<N-1:B+W>的位值,以及(iii)第三前缀值,该第三前缀值具有与该相应第一源数据元素的位<A+W-1:A>的符号扩展相对应的位值;以及

[0014] (c)若B＞0,则为具有与后缀值相对应的位值的位<B-1:0>,根据该控制值选择该后缀值为以下中的一者:(i)第一后缀值,该第一后缀值包含各自具有零值的位,以及(ii)第二后缀值,该第二后缀值具有该相应第二源数据元素的位<B-1:0>的位值;

[0015] 其中,该位域操纵指令具有位域插入形式,其中该源位位置A＝0而该结果位位置B＞0,并且具有位域提取形式,其中该源位位置A≥0而该结果位位置B＝0;以及

[0016] 该控制值指定用于确定该位域宽度W、该源位位置A及该结果位位置B的最高有效位位置S和旋转参数R;

[0017] 若S≥R,则该源位位置A＝R且该结果位位置B＝0,该位域宽度W＝(S+1)-R,且该位域操纵指令具有该位域提取形式;以及

[0018] 若S＜R,则该源位位置A＝0且该结果位位置B＝N-R,该位域宽度W＝S+1,且该位域操纵指令具有该位域插入形式。

[0019] 处理装置被提供有处理电路及指令译码器,该指令译码器响应于位域操纵指令以控制处理电路执行位域操纵操作。位域操纵指令至少识别第一源数据值及第二源数据值,该第一源数据值包含至少一个第一源数据元素,该第二源数据值包含至少一个第二源数据元素。响应于位域操纵指令,处理电路被控制来产生结果数据值,该结果数据值包含至少一个结果数据元素。每一个第一源数据元素、第二源数据元素及结果数据元素具有N位<N-1:0>。

[0020] 每一个N位结果数据元素包括W位<B+W-1:B>,所述W位<B+W-1:B>具有与第一源数据值的相应第一源数据元素的位<A+W-1:A>相对应的位值。因此,每一个结果数据元素包括自相应第一源数据元素提取的W位位域。源位位置A指示第一源数据元素内位域的最低有效位的位置,结果位位置B指示结果数据元素内位域的最低有效位,且位域宽度W指示位域所包含的位数目。量A、量B及量W位于1≤W≤N,0≤A≤N-W及0≤B≤N-W的范围内,且由位域操纵指令内的控制值识别。控制值可直接识别A、B及W,或使用可用于推导A、B及W的任何参数组间接识别A、B及W。应注意,控制值不必为位域操纵指令编码内的一组连续的位,但亦可包含分布于指令编码中各处的两组或更多组的位。

[0021] 若控制值指示B+W＜N,则每一个结果数据元素将包括前缀部分,该前缀部分包含位比结果数据元素内位域的最高有效位<B+W-1>更有效的<N-1:B+W>。本技术为设定前缀部分的位值提供了不同的选择。位域操纵指令的控制值包括指示何种类型的前缀部分将用于特定位域操纵的信息。根据控制值,前缀值被选择为第一前缀值、第二前缀值及第三前缀值中的一个。

[0022] 第一前缀值包含各自具有零值的位。当选定第一前缀值时,每一个结果数据元素含有从该相应源数据元素提取的位域,其中比该位域更有效的任何位被设定为0。此举可用于隔离源数据元素的位域,以便可以与源数据元素的其他部分相分离地来处理该位域的值。

[0023] 第二前缀值具有该相应第二源数据元素的位<N-1:B+W>的位值。因此，当选定第二前缀值时，则位域操纵产生这样的结果，该结果等效于将第一源数据元素的位域插入第二源数据元素内，其中第二源数据元素的任何更有效的位保持无变化。此举可用于将两个或两个以上数据值的各部分组合成为单个数据值。例如，通过使用第二前缀形式的位域操纵指令将与另外两种成分相对应的位域插入含有RGB成分中的一种成分的数据值内，可从分离的RGB成分值组装成组合RGB色值。

[0024] 第三前缀值具有与相应第一源数据元素的位<A+W-1:A>的符号扩展相对应的位值。此举适用于表示带符号值的位域，因为当符号扩展被包括于结果数据元素内时符号扩展保存第一源数据元素的位域的符号。例如，若自第一源数据元素提取的位域<A+W-1:A>表示负数，则由第三前缀值提供的符号扩展维持结果数据元素内的位域的负号。

[0025] 通过选择第一前缀值、第二前缀值及第三前缀值中的合适的一个前缀值，可通过相同的位域操纵指令来对不同种类的位域操纵编码。由于单个位域操纵指令可提供多个不同的操纵操作，故可高效率地使用指令集的编码空间。熟习此项技术者将了解：用于实施位域操纵指令的硬件可具有各式各样的不同形式，且处理电路及指令译码器可包含用于产生具有上述形式的至少一个结果数据元素的结果值的任何种类的硬件。

[0026] 取决于用于数据值的符号表示的类型，第三前缀值的符号扩展位可具有不同形式。然而，在一个示例中，第三前缀值可包含各自具有相应第一源数据元素的位<A+W-1>的位值的位。在许多二进制带符号数表示中，指示数据值为正或为负的符号位为该数据值的最高有效位。自第一源数据元素提取的位域的最高有效位为位<A+W-1>，且因此通过在第三前缀值的各位中复制该位，当将该位域插入结果数据值内时，被提取位域的符号得到维持。

[0027] 若控制值将结果位位置B定义为大于0，则结果数据元素具有这样的后缀部分，该后缀部分包括不及位域的最低有效位<B>有效的至少一个位<B-1:0>。根据控制值，后缀值可被选定为第一后缀值及第二后缀值中的一个，该第一后缀值包含各自具有零值的位，该第二后缀值具有相应第二源数据元素的位<B-1:0>的位值。通过选择第一后缀值，可将位域在结果数据元素内与具有零值的较低有效位隔离。通过使用第二后缀值，位域操纵将第一源数据元素的位域有效地插入第二源数据元素内，其中第二源数据元素的较低有效位无变化。

[0028] 尽管在随后所述示例中，描述了指令编码，其中当选定第二前缀值时则选定了第二后缀值，且当选定第一前缀值及第三前缀值中的一个时则选定了第一后缀值，但应了解，亦可使用第一、第二及第三前缀值中的一个与第一及第二后缀值中的一个的任何任意组合。

[0029] 该数据处理装置可包含复数个储存位置，被配置为储存供处理电路处理的数据值；

[0030] 其中该位域操纵指令至少识别用于储存第一源数据值的第一源储存位置及目的地储存位置；并且

[0031] 该指令译码器响应于位域操纵指令，产生用于控制处理电路将该结果数据值储存于目的地储存位置中的控制信号。

[0032] 在位域操纵指令的一个示例中，该指令至少识别第一源储存位置及目的地储存位置，该第一源储存位置用于储存第一源数据值，该目的地储存位置用于储存所产生的结果

数据值。例如，储存位置可为寄存器、存储器位置或用于储存供处理电路处理的数据的任何其他数据储存器。

[0033] 可选地，位域操纵指令可指定用于储存第二源数据值的第二源储存位置，或第二源数据值可为在该指令内立即识别的瞬时值。

[0034] 或者，在一个示例中，第二源数据值可为在执行位域操纵指令之前存在于目的地储存位置中的数据值。在此情况下，结果数据值覆写目的地储存位置内的第二源数据值。若位域操纵指令的控制值使得第二前缀值及第二后缀值被选定，则位域操纵的结果为：将来自第一源数据值的位域插入目的地储存位置内，其中目的地储存位置的其他位保持无变化。

[0035] 一些处理装置可允许储存位置(例如，寄存器)作为复数个不同储存位置大小的储存位置进行存取。因此，位域操纵指令的控制值可包括用于识别第一源储存位置及目的地储存位置的储存位置大小的信息。可以以不同方式来实现储存位置。在一个示例中，可能存在实体上不同的储存位置，所述储存位置具有不同的大小(例如，一组64位储存位置及另一组32位储存位置)。或者，一组共同的储存位置可以以不同储存位置大小进行存取。例如，相同的64位储存位置可经受64位数据存取及32位数据存取。在64位储存位置的32位数据存取期间，仅可读取储存位置的位中的32个位，或者替代地，可读取储存位置的所有64个位但在执行数据值的任何进一步处理之前可摒弃所述位中的32个位。同样地，当将32位数据值写入64位储存位置时，其他32个位可被设定为一些预定值，或可设定为32位数据的符号扩展，或者保持无变化。

[0036] 在一些实施例中，包括于第一源数据值、第二源数据值及结果数据值内的一个或多个数据元素可具有数据元素大小N，该数据元素大小N选自复数个不同的数据元素大小。在此情况下，于是位域操纵指令的控制值亦可包括用于直接或间接地识别数据元素大小N的信息。提供可变的数据元素大小例如可适用于单个指令多个数据(single instruction multiple data;SIMD)处理操作，其中位域操纵被并行应用于第一源数据值、第二源数据值及结果数据值内的多组相应的数据元素。

[0037] 由位域操纵指令的控制值来识别位位置A及位位置B，位位置A及位位置B指示第一源数据元素及结果数据元素内位域的位置。在位域操纵指令的一个示例中，控制值可识别A及B的任何任意值，使得可将来自源数据元素内的任何位置的位域复制至结果数据元素内的任何位置。

[0038] 然而，在一个示例中，位域操纵指令具有这样的编码，该编码使得源位位置A及结果位位置B中的一个具有零值。本技术认为：对于最通常希望类型的位域操纵操作而言，源位位置A及结果位位置B中的至少一个为0。通过将A及B中的一者设定为等于0，随后A及B中的仅非零的一个需要由控制值识别，且因此指令编码变得更高效。在该指令的位域插入形式中，A＝0而B＞0，以使得位于源数据元素的最低有效部分<W-1:0>处的位域被插入结果数据元素的任意部分<B+W-1:B>中。在该指令的位域提取形式中，A≥0而B＝0，以使得位域从源数据元素的任意部分<A+W-1:A>被提取，且被复制至结果数据元素的最低有效部分<W-1:0>。指令的提取形式及插入形式满足最常用形式的位域操纵。然而，若需要希望A及B两者为非零值的位域操纵，则仍可使用如下两个位域操纵指令执行此操作：位域提取形式的指令，该指令用于自源值的任意位置A提取源位域且将位域复制至暂时储存位置的最低有效部

分;和位域插入形式的指令,该指令自暂时储存位置的最低有效部分取出位域且将该位域插入结果值内的任意位置B内。

[0039] 在一个示例中,该装置可被配置为使该控制值指定最高有效位位置S及元素旋转参数R,以确定该位域宽度W、该源位位置A及该结果位位置B;

[0040] 若S≥R,则该源位位置A=R且该结果位位置B=0,该位域宽度W=(S+1)-R且该位域操纵指令具有该位域提取形式;以及

[0041] 若S<R,则该源位位置A=0且该结果位位置B=N-R,该位域宽度W=S+1,且该位域操纵指令具有该位域插入形式。

[0042] 对位域操纵指令的该编码特别高效,因为两个参数S及R足以至少识别:(a)位域操纵指令为位域提取形式还是位域插入形式,(b)源位位置A,该源位位置A指示第一源数据元素内位域的位置,(c)结果位位置B,该结果位位置B指示结果数据元素内位域的位置;以及(d)位域宽度W,该位域宽度W指示由该位域包含的位的数目。

[0043] 对于位域插入形式及位域提取形式两者而言,最高有效位位置S表示在第一源数据元素内位域的最高有效位的位位置,且元素旋转参数R表示如下的位位置数目,若源位位置A被移动至结果位位置B,则每一个源数据元素将向右被旋转该位位置数目。对位域插入形式而言,R表示第一源数据元素内位域的最低有效位的位位置,而对位域提取形式而言,(N-R)值识别出结果数据元素内位域的最低有效位位置(其中N为数据元素大小)。

[0044] 尽管元素旋转参数R表示位位置的数目(若源位位置A移动至结果位位置B,则每一个源数据元素将向右旋转该数目),但并非绝对必要在从第一源数据元素产生结果数据元素时实际执行向右旋转。例如,等效的左旋转可代替右旋转来使用,或可在不实际执行旋转的情况下产生结果数据元素。

[0045] 在一个示例中,最高有效位位置S可包含位域符号位参数S。若第一源数据元素为带符号数据值,则最高有效位位置S(除识别上述项目(a)至项目(d)外)亦识别第一源数据元素内的符号位的位置。

[0046] 该装置可被布置为使得该控制值包括第一部分及第二部分,第一部分及第二部分各自包含可变数目的位;并且

[0047] 该指令译码器及该处理电路中的至少一者响应于位域操纵指令,确定由该控制值的该第一部分包含的位的数目,且基于由该第一部分包含的位的数目确定:

[0048] (a)该至少一个第一源数据元素、该至少一个第二源数据元素及该至少一个结果数据元素的数据元素大小N;以及

[0049] (b)该第二部分中的哪些位指示该最高有效位位置S及该旋转参数R。

[0050] 数据元素大小N愈大,则R及S的可能值愈大。因此,表示R及S所需的位数目将视针对特定操作而选定的数据元素大小N而变化。尽管可能在控制值内指派足以识别R及S的最大可能值的固定数目的位,但可通过提供具有各自有可变长度的第一部分及第二部分的控制值来达成更高效的编码。视所使用的数据元素大小N,可将控制值的位可变地指派至第一部分或第二部分。通过检测长度可变的第一部分的大小,装置可识别数据元素大小N且该装置可识别第二部分中的哪些位表示最高有效位位置S及旋转参数R。

[0051] 在一些实施例中,可按照与第一及第二源数据值内的相应第一源数据元素及第二源数据元素相同的次序布置结果数据元素,来生成结果数据值。

[0052] 然而，可布置其他实施例为使得，若该第一源数据值包含复数个第一源数据元素、该第二源数据值包含复数个第二源数据元素且该结果数据值包含复数个结果数据元素，则该控制值包括数据元素排序信息，该数据元素排序信息用于指示将该复数个结果数据元素布置于该结果数据值内所使用的次序。

[0053] 因此，在需要时，位域操纵指令亦可用以实施数据元素重新排序。例如，可基于如下第一源数据值及如下第二源数据值产生如下结果数据值，该结果数据值包含两个结果数据元素A及B，该第一源数据值包含相应第一源数据元素A′及B′，该第二源数据值包含相应第二源数据元素A″及B″。随后，由控制值指示的数据元素排序信息可指定是以次序AB(对应于源数据值中相应源数据元素的次序)还是以相反的次序BA布置结果数据元素。

[0054] 应注意，在需要时，位域操纵指令可用以实施数据元素在数据值内的重新排序，即使不对该数据值执行位域操纵。可通过将位域宽度W设定为与数据元素大小N相同来实现此举，以使位域操纵操作具有将整个第一源数据值复制至结果数据值的效应，其中依据数据元素排序信息将结果数据值内的数据元素重新排序。因此，本技术提供不仅可实施位域操纵而且可实施数据元素的重新排序的通用的指令。

[0055] 在一个示例性实施例中，该装置可配置为使得，若该第一源数据值包含复数个第一源数据元素，该第二源数据值包含复数个第二源数据元素且该结果数据值包含复数个结果数据元素，则该结果数据值等效于第一数据值，该第一数据值通过以下获得：

[0056] (a)产生中间值，该中间值包含与该第一源数据值中的相应第一源数据元素的次序及该第二源数据值中的相应第二源数据元素的次序相对应地被排序的所述结果数据元素，以及

[0057] (b)对该中间值内的所述结果数据元素执行至少一次重新排序迭代以产生该第一数据值；

[0058] 每一个重新排序迭代包含：判断该数据元素排序信息的相应位是否具有预定值，且若该数据元素排序信息的该相应位具有预定值，则交换该中间值内的各对位组(pairs of groups of bits)。

[0059] 在一个示例中，中间值内被交换对的位组包含相邻对的位组。

[0060] 实施数据元素重新排序的一种方式为产生在形式上等效于在以下情况下产生的第一数据值的结果数据值：(a)使用如上所述的位域操纵操作，产生中间值，其中按与第一源数据值及第二源数据值内的相应第一源数据元素及第二源数据元素相同的次序将从位域操纵产生的结果数据元素排序，及(b)将一系列一次或多次重新排序迭代应用于中间值以产生第一数据值，每一次重新排序迭代包含以下步骤：若数据元素排序信息的相应位具有预定值，则交换中间值内的多对位组。

[0061] 注意，实际上不必通过执行如上所述的步骤(a)及步骤(b)来产生结果数据值。在一些实施方案中，处理电路可在单个操作中产生最终结果值，其中位域操纵操作基于第一源数据值/第二源数据值被执行，且结果数据元素也按照结果数据值内的所希望次序被排序。因此，不必通过处理电路产生上述中间值，或处理电路不必实际上执行重新排序迭代。最终结果数据值具有等效于第一数据值的值，将通过对中间值执行这样的重新排序迭代来产生该第一数据值。

[0062] 尽管在一些实施例中，重新排序迭代可交换具有任意数目的位的数对位组，但若

位组所包含的位数目为数据元素大小N的倍数，则可简化用于产生结果数据值的数据元素排序信息的编码及硬件配置。例如，重新排序迭代可交换数对单个数据元素或交换数对两个、四个或更多数据元素的组。

[0063] 在一个示例中，位组(the groups of bits)可包含针对至少一个重新排序迭代中的不同重新排序迭代的不同位数目。用于交换不同组大小的位组(或数据元素的组)的一系列重新排序迭代使得能够对将被提供在结果数据值内的数据元素进行各式各样可能的布置。

[0064] 装置可被配置为使该控制值包括第一部分及第二部分，第一部分及第二部分各自包含可变数目的位；并且

[0065] 该指令译码器及该处理电路中的至少一者响应于该位域操纵指令确定该控制值的该第一部分所包含的位数目，且基于由该第一部分包含的位数目确定：

[0066] (a)该至少一个第一源数据元素、至少一个第二源数据元素及该至少一个结果数据元素的数据元素大小N；以及

[0067] (b)第二部分中的哪些位指示该数据元素排序信息。

[0068] 以与上述参数S及R相似的方式，可使用各自具有可变数目的位的第一部分及第二部分为数据元素排序信息编码。此举有用，因为数据元素排序信息通常将需要依据有多少数据元素存在于数据值内而定的位数(数据元素的数目愈大，则可能的重新排序排列的数目愈大)。数据元素的数目取决于数据元素大小N，且因此需要用来表示数据元素排序信息的位数可与数据元素大小N相反地变化(数据元素大小愈大，可能的重新排序排列的数目愈小)。因此，针对位域参数S及参数R的如上所述的控制值的编码方案可类似地用于以有高效方式表示数据元素排序信息。

[0069] 从又一方面来看，本发明提供一种用于处理装置执行处理操作的数据处理方法，该方法包含：

[0070] 响应于位域操纵指令，至少识别第一源数据值、第二源数据值及控制值，该第一源数据值包含各自具有N位<N-1:0>的至少一个第一源数据元素，该第二源数据值包含各自具有N位<N-1:0>的至少一个第二源数据元素，产生用于控制该处理装置产生结果数据值的控制信号，该结果数据值包含各自具有N位<N-1:0>的至少一个结果数据元素，每一个结果数据元素与相应第一源数据元素及相应第二源数据元素相对应；其中：

[0071] 该控制值包括用于指示位域宽度W、源位位置A及结果位位置B的信息，其中1≤W≤N，0≤A≤N-W及0≤B≤N-W，且每一个结果数据元素包含：

[0072] (a)具有与该相应第一源数据元素的位<A+W-1:A>相对应的位值的位<B+W-1:B>；

[0073] (b)若B+W＜N，则为具有与前缀值相对应的位值的位<N-1:B+W>，根据该控制值选择该前缀值为以下中的一者：(i)第一前缀值，该第一前缀值包含各自具有零值的位，(ii)第二前缀值，该第二前缀值具有该相应第二源数据元素的位<N-1:B+W>的位值，以及(iii)第三前缀值，该第三前缀值具有与该相应第一源数据元素的位<A+W-1:A>的符号扩展相对应的位值；以及

[0074] (c)若B＞0，则为具有与后缀值相对应的位值的位<B-1:0>，根据该控制值选择该后缀值为以下中的一者：(i)第一后缀值，该第一后缀值包含各自具有零值的位，以及(ii)第二后缀值，该第二后缀值具有该相应第二源数据元素的位<B-1:0>的位值；

[0075] 其中，该位域操纵指令具有位域插入形式，其中该源位位置A＝0而该结果位位置B＞0，并且具有位域提取形式，其中该源位位置A≥0而该结果位位置B＝0；以及

[0076] 该控制值指定用于确定该位域宽度W、该源位位置A及该结果位位置B的最高有效位位置S和旋转参数R；

[0077] 若S≥R，则该源位位置A＝R且该结果位位置B＝0，该位域宽度W＝(S+1)-R，且该位域操纵指令具有该位域提取形式；以及

[0078] 若S＜R，则该源位位置A＝0且该结果位位置B＝N-R，该位域宽度W＝S+1，且该位域操纵指令具有该位域插入形式。

[0079] 从另一方面来看，本发明提供一种数据处理装置，该数据处理装置包含：

[0080] 处理电路，该处理电路被配置为执行处理操作；

[0081] 指令译码器，该指令译码器响应于程序指令以产生用于控制该处理电路执行所述处理操作的控制信号；其中：

[0082] 所述程序指令包括至少一个指令，该至少一个指令指定一控制值，该控制值具有第一部分及第二部分，该第一部分用于指示选自复数个数据大小的选定数据大小，该第二部分用于指示至少一个控制参数，该至少一个控制参数具有根据该选定数据大小而变化的位数目，该第一部分及该第二部分各自具有可变数目的位；以及

[0083] 该指令译码器响应于该至少一个指令产生用于控制该处理电路根据该选定数据大小及该至少一个控制参数执行相应处理操作的控制信号；

[0084] 其中当处理该至少一个指令时，该指令译码器及该处理电路中的至少一者被配置为识别由该控制值的该第一部分包含的位数目，且根据由该第一部分包含的该位数目来识别：(a)该选定数据大小，以及(b)该控制值中的哪些位形成用于指示该至少一个控制参数的该第二部分。

[0085] 如上所述，位域操纵指令可被与选定数据元素大小相关联，该选定数据元素大小指示将经受位域操纵的数据元素的大小。用于控制位域操纵操作的控制参数可具有视数据元素大小而定的可变位数，且因此如上所述，使用可变长度的第一部分及第二部分的编码方案可被用于以高效的方式识别这些控制参数。

[0086] 该类型的控制值的编码亦可用于与如下处理操作相关联的其他种类的指令，该处理操作取决于选自复数个数据大小的数据大小及至少一个控制参数，该至少一个控制参数具有视选定数据大小而定的位数。对此等指令而言，指令可指定控制值，该控制值具有第一部分及第二部分，该第一部分及该第二部分各自具有可变数目的位。通过识别第一部分所包含的位数目，指令译码器及处理电路中的至少一者可确定选定数据大小且识别控制值中的哪些位对应于表示该至少一个控制参数的第二部分。这种形式的指令编码可应用于许多类型的指令，且提供了对控制参数编码的高效方式，所述控制参数的位数目视选定数据大小而变化。

[0087] 注意，术语“部分”不一定表示控制值内的位的连续部分，该术语亦可代表控制值的任何任意位组(即使这些位不具有相邻的位位置)。

[0088] 在可用指令集包括复数个不同种类的指令(每一种指令使用控制值的该常用编码格式来识别与该指令相关联的控制参数及数据大小)的实施例中，可使指令译码器及/或处理电路更高效，因为用于对控制值译码的硬件的一部分可再用于不同种类的指令。

[0089] 若随着第二部分中的位数目视选定数据大小而减小，第一部分的位数目视选定数据大小而增大，则该技术尤其有用。通过随着第二部分中的位数目减小而增大第一部分的位数目且反之亦然，则第一部分可使用控制值中第二部分不需要的任何位来指示选定数据大小。因此，控制值的总大小可保持不变而不管选定数据大小如何，其中根据所使用的特定数据大小适当地将控制值的不同位分配给第一部分或第二部分。

[0090] 在一个示例中，控制值的第一部分可被编码为使得对于复数个数据大小的至少一子集而言，该第一部分至少包含第一位及X个剩余位，该第一位具有第一状态，所述剩余位具有第二状态，其中X为大于0或等于0的可变整数；并且

[0091] 该指令译码器及该处理电路中的该至少一者被配置为：根据该控制值的预定部分内该第一位的位位置，识别该控制值的该第一部分所包含的位数目。

[0092] 在该示例中，第一部分包括具有第一状态(例如，“0”状态或“1”状态)的至少一个位。第一部分的零个、一个或多个剩余位具有第二状态(例如，“0”状态及“1”状态中的另一状态)。在一个示例中，具有第二状态的第一部分的零个、一个或多个位可有效地用作用于填充控制值的如下位位置的“填补(padding)”位，第二部分不需要所述位位置来指示针对给定数据大小的至少一个控制参数。视具有第二状态的位数目而定，具有第一状态的位的位置可变化。然后，可基于具有第一状态的位的位位置来识别数据大小。

[0093] 基于具有第一状态的第一位的位位置来检测第一部分的大小(及因此检测选定数据大小)的技术尤其有用，因为在许多实施方案中，可用数据大小将以2的幂增大，且因此具有第一状态的位的位位置通常可被与用于该选定数据大小的2的特定幂数相关。

[0094] 识别具有第一状态的第一部分的位的位位置的技术不必为用以识别数据元素大小的唯一技术。亦可能存在至少一个额外位，该至少一个额外位提供用于识别该选定数据大小的其他信息。

[0095] 尽管可能使用本控制值编码技术来仅表示单个控制参数，但在存在复数个控制参数时，该技术尤其有用。在此情况下，于是当处理至少一个指令时，指令译码器或处理电路可被配置为：根据第一部分所包含的位数目，识别第二部分中的哪些位指示复数个控制参数中的每一个。

[0096] 当存在复数个控制参数时，然后若控制参数至少包括如下第一控制参数及如下第二控制参数，则本编码技术尤其有用，该第一控制参数的位数目随选定数据大小增大而增大，该第二控制参数的位数目随选定数据大小增大而减小。在此情况下，然后第二部分可包括指示第一控制参数及第二控制参数的子部分，其中根据选定数据大小将该子部分的位分配给第一控制参数或第二控制参数。将位数目随选定数据大小增大而增大的控制参数与位数目随选定数据大小增大而减小的另一控制参数配对可产生控制值的高效率的编码，因为可将不需要用于指示针对给定数据大小的第一控制参数的位再分配用于指示第二控制参数，且反之亦然。

[0097] 如上所述，可将本编码技术应用于许多不同类型的指令。该技术尤其适用的一类指令为单个指令多个数据(SIMD)指令，该指令识别具有至少一个源数据元素的源数据值。对至少一种SIMD指令而言，选定数据大小可指示至少一个源数据元素的数据元素大小，且相应处理操作可包含以下步骤：对源数据值内的每一个源数据元素并行执行选定处理操作。用于控制相应处理操作的控制参数可具有不同数据元素大小的不同的容许范围，且因

此该控制参数的位数目可随数据元素大小而变化。因此，本编码技术可用于对控制值内的选定数据元素大小及可变长度控制参数两者高效地编码。

[0098] 本控制值编码技术可应用于的SIMD指令的一个特定示例为位域操纵指令，对该位域操纵指令而言，该相应处理操作包含以下步骤：产生包含至少一个结果数据元素的结果数据值，每一个结果数据元素对应于该源数据值的一相应源数据元素；

[0099] 每一个结果数据元素包含结果位域，该结果位域具有与该相应源数据元素内连续位的源位域的位值相对应的位值；以及

[0100] 该至少一个控制参数指示由该源位域及该结果位域所包含的位数目、该相应源数据元素内的该源位域的位置及该结果数据元素内的该结果位域的位置。

[0101] 因此，控制值包括第一部分及第二部分，该第一部分的位数目识别位域操纵指令的数据元素大小，该第二部分识别指示位域宽度、源数据元素内位域的位置及结果数据元素内结果位域的位置的控制参数。数据元素大小愈大，则位域宽度以及源数据元素及结果数据元素内位域位置的可能值愈大，且因此指示这些参数所需要的位数目愈大。因此，本控制值编码技术适用于指示这些参数。

[0102] 位域操纵指令亦可指定数据元素排序参数，该数据元素排序参数识别将结果数据元素布置于结果值内所使用的次序。类似地，使用具有可变大小的第一部分及第二部分的本编码技术可用以识别数据元素排序参数。

[0103] 本编码技术可应用于的另一种类型的指令为按位(bitwise)逻辑指令，该按位逻辑指令识别源数据值且对该按位逻辑指令而言，相应处理操作包含以下步骤：产生如下这样的结果数据值，该结果数据值等效于将逻辑运算应用至源数据值及基于控制值确定的第二数据值的结果。

[0104] 可将第二数据值视为掩码(mask)值，使用逻辑运算将该掩码值与源数据值组合。在此情况下，选定数据大小指示由第二数据值内重复模式的位所包含的位数目，且至少一个控制参数指示用于识别重复模式的位的位值的信息。因此，在此情况下，选定数据大小不必为数据元素大小，但指示第二数据值内重复模式的位的大小。重复模式的位的大小愈大，则控制参数中识别这些位的位值所需的位数目愈大，且因此控制参数将具有随选定数据大小而变化的位数目。因此，本编码技术适用于指示这样的参数。

[0105] 与按位逻辑指令相关联的逻辑运算可包括若干不同类型的逻辑运算。例如，逻辑运算可包含与(AND)运算、或(OR)运算及异或(XOR)运算中的一者。

[0106] 注意，按位逻辑指令不一定控制处理电路实际产生含有重复模式的位的第二数据值且将逻辑运算应用至源数据值及第二数据值。可能地，该处理电路可包括用于以单个组合运算产生结果数据值的硬件，该结果数据值等效于在以下情况下会获得的结果：第二数据值被生成且使用逻辑运算将该第二数据值与源数据值组合。该第二数据值不必实际地存在于该运算的任何阶段。

[0107] 从另一方面来看，本发明提供一种用于处理装置响应于程序指令执行处理操作的数据处理方法，该数据处理方法包含以下步骤：

[0108] 接收至少一个指令，该至少一个指令指定一控制值，该控制值具有第一部分及第二部分，该第一部分用于指示选自复数个数据大小的选定数据大小，该第二部分用于指示至少一个控制参数，该至少一个控制参数具有根据该选定数据大小而变化的位数目，该第

一部分及该第二部分各自具有可变数目的位；

[0109] 响应于该至少一个指令，识别由该控制值的该第一部分包含的位数目；

[0110] 根据由该第一部分包含的位数目，识别：(a)该选定数据大小，以及(b)该控制值中的哪些位形成用于指示该至少一个控制参数的该第二部分；以及

[0111] 控制该处理装置根据该选定数据大小及该至少一个控制参数执行相应处理操作。

[0112] 本发明亦提供一种虚拟机，该虚拟机由计算机程序提供，当由计算机执行时，该计算机程序根据依据上述任一示例的数据处理装置提供指令执行环境。例如，虚拟机可对含有上述指令中的一个指令的程序与用于执行该指令的硬件的互动进行建模。用于执行虚拟机的主计算机自身不必含有能够执行指令的译码及处理硬件，但该主计算机包括能够执行虚拟机的足够的处理资源，该虚拟机仿真该指令的处理。

[0113] 从结合附图阅读的说明性实施例的以下详细描述，本发明的上述及其他目的、特征及优点将清楚。

附图说明

[0114] 图1示意性图示出数据处理装置；

[0115] 图2示出位域操纵指令的示例编码；

[0116] 图3图示出位域操纵操作的示例；

[0117] 图4示出通过应用位域操纵而自源数据元素产生结果数据元素的示例；

[0118] 图5示出位域操纵指令内的控制域的示例编码；

[0119] 图6图示出位域插入形式的位域操纵指令及位域提取形式的位域操纵指令的示例；

[0120] 图7示出根据位域操纵指令内的控制值选择结果数据元素的不同前缀部分及后缀部分的示例；

[0121] 图8A图示出根据控制值内经编码的重新排序信息将结果数据值内的数据元素重新排序的示例；

[0122] 图8B图示出将数据元素重新排序的第二示例；

[0123] 图9图示出处理位域操纵指令的方法；

[0124] 图10图示出对控制域译码以获得控制参数的值的方法；

[0125] 图11图示出按位逻辑指令的示例编码，该按位逻辑指令与位域操纵指令共享共同的控制域编码；

[0126] 图12图示出按位逻辑指令的控制域的示例编码；

[0127] 图13图示出掩码值的示例，使用逻辑运算将该掩码值与源数据值组合；

[0128] 图14图示出可在按位逻辑指令的控制下执行的不同种类的逻辑运算；

[0129] 图15图示出处理按位逻辑指令的示例方法；以及

[0130] 图16图示出虚拟机实施方案的示例。

具体实施方式

[0131] 图1示意性图示出数据处理装置2，该数据处理装置2包含处理电路4、指令译码器6、寄存器组8及存储器系统10。存储器系统10包括一个或多个高速缓存或存储器器件。处理

电路4包括若干处理组件，诸如加法器12、乘法器14及移位器16。当处理数据时，指令译码器6响应于程序指令，产生用于控制处理电路4处理数据(所述数据储存于寄存器8中)且将处理结果储存于寄存器8中的控制信号。在指令译码器6的控制下，亦可控制数据处理装置2在寄存器8与存储器系统10之间传送数据。

[0132] 可使用多个不同的寄存器存取大小来存取寄存器组8。若寄存器包含(例如)64位，则寄存器可经受(例如)64位存取或32位存取。经由指令译码器6译码的指令可包括信息，该信息指定待用于给定处理操作的选定寄存器存取大小。

[0133] 尽管在图1的示例中，处理电路4图示出为处理储存于寄存器8中的数据且将数据处理结果储存回至寄存器8，但应了解，任何其他种类的储存位置可代替寄存器8来使用。将了解，数据处理装置2及处理电路4通常可包括未图示于图1中的其他组件。

[0134] 图2示出位域操纵指令BF的示例编码，该位域操纵指令BF用于控制处理装置2执行位域操纵操作。由位于该指令的位<28:23>处的作业码(opcode)来识别位域操纵指令，且该位域操纵指令包括以下域：

[0135] • Rn：源寄存器域，该源寄存器域识别寄存器组8中的储存第一源数据值src1的寄存器。

[0136] • Rd：目的地寄存器域，该目的地寄存器域识别寄存器组8中的处理结果被储存至的目的地寄存器。目的地寄存器域Rd亦识别第二源数据值src2，该第二源数据值src2为在执行位域操纵指令之前储存于目的地寄存器中的值。

[0137] • sf：寄存器存取大小域，该寄存器存取大小域识别待用于源寄存器及目的地寄存器的选定存取大小M。在图2的示例中，寄存器大小域sf包含单个大小指示位，其中大小指示位的第一状态指示第一存取大小(例如，64位)且大小域的第二状态指示第二存取大小(例如，32位)。然而，在需要时，可通过将位域操纵指令编码的更多位分配至寄存器存取大小域sf，来提供大于两个的不同存取大小。

[0138] • opc：子作业码域，该子作业码域指示用于控制将由处理电路4执行的位域操纵类型的控制值。下文参考图7描述了子作业码域的示例编码。

[0139] • Control13：控制域，该控制域识别：

[0140] o第一源数据值src1、第二源数据值src2及将在位域操纵指令的处理期间产生的结果数据值的数据元素的数据元素大小N；以及

[0141] o用于控制将由处理电路4执行的位域操纵操作的各方面的若干其他控制参数。

[0142] 下文将结合图5描述control13域的编码的示例。

[0143] 亦可自sf域及control13域推导出数据元素的数目。存在于第一源数据值src1及第二源数据值src2以及结果数据值的每一个内的数据元素的数目等于M/N，其中M为选定寄存器存取大小，而N为数据元素大小。

[0144] 可将sf域、opc域及control13域总体地视为形成用于控制位域操纵操作的各方面的控制值。

[0145] 尽管图2的示例编码图示出将第二源数据值src2识别为在执行位域操纵指令之前储存于目的地寄存器中的值，但将了解，在其他实施例中，可提供单独的寄存器域来识别储存第二源数据值src2的寄存器(而非目的地寄存器)。

[0146] 图3图示出位域操纵操作的示例。第一源值src1含有M/N个数据元素，每个数据元

素具有N位，该第一源值src1为在执行位域操纵指令之前储存于寄存器Rn中的M位的值。类似地，储存于目的地寄存器Rd中的M位的第二源值src2亦包括各自具有N位的M/N个数据元素，所述数据元素。

[0147] 响应于位域操纵指令，处理电路4产生结果数据值，该结果数据值具有M/N个结果数据元素，所述M/N个结果数据元素对应于源值src1、src2的数据元素。每一个结果数据元素包括W个位，所述W个位的值对应于第一源数据值src1内W个位的位域(bf)。可基于位域操纵指令的控制值来控制第一源数据值src1内位域的位置及结果数据值内位域的位置。对每一个结果数据元素而言，不如所插入位域有效的任何位采用后缀值s的值，而比该位域更有效的结果数据元素的任何位采用前缀值p的值。将结果数据元素储存于目的地寄存器Rd中。

[0148] 因此，位域操纵指令的效应为：自第一源值src1的每一个数据元素内的给定位置提取位域，且将每一个经提取的位域插入结果值的相应数据元素内，其中目的地寄存器的其他位(若有的话)采用前缀值及后缀值。

[0149] 图4图示出可定量地定义位域操纵操作的方式的示例。图4图示出被应用于单个源数据元素以产生相应结果数据元素的位域操纵操作的示例。若源数据值及结果数据值具有两个或两个以上数据元素，则可将相同操作应用于每一个源数据元素以产生相应结果数据元素。

[0150] 指令译码器6响应于位域操纵指令产生结果数据元素，其中结果数据元素的位<B+W-1:B>采用位域bf的值，该位域bf包含第一源值src1的相应源数据元素的位<A+W-1:A>。参数A指示将从第一源数据值src1提取的位域的开始位位置，而参数B指示位域将被放置于结果数据元素内的开始位位置。W指示位域bf所包含的位数目。A、B及W具有1≤W≤N，0≤A≤N-W及0≤B≤N-W范围内的任何整数值，其中N为该数据元素所包含的位数目。

[0151] 图4图示出位域操纵的一般形式，其中位域位置值A及B可分别采用源数据元素及结果数据元素内的任何位位置的值。将A、B及W直接编码于位域操纵指令的控制值内是可能的。

[0152] 然而，在具体实施例中，如图5及图6所示，control3域的控制编码被用以间接地识别A、B及W。

[0153] 图5图示出位域操纵指令的control3域的示例编码。control3域具有13位<12:0>，包括第一部分30及第二部分32，该第一部分30及该第二部分32的每个部分具有可变数目的位。control3域识别数据元素大小N、旋转参数R、最高有效位位置S及反转信息V。旋转参数R及最高有效位位置S确定：将从源数据元素src1提取的位域的位置及大小，及将位域插入结果数据元素内的位置，下文将结合图6阐释此举。反转信息V定义将数据元素布置于结果数据元素内的次序，下文将结合图7阐释此举。

[0154] control3域的第一部分30包含可变数目的位。在该示例中，第一部分30包含control3域的位<12>，以及control3域的位<5:0>中的零个、一个或多个位。通过检测第一部分30所包含的位数目，可识别数据元素大小N。

[0155] control3域的第二部分32具有可变数目的位，所述位表示旋转参数R、最高有效位位置S及反转信息V。control3域中的哪些位表示哪一个参数视用于给定指令的特定数据元素大小N而定。指令译码器6或处理电路4基于第一部分30中所识别的位数目，来识别第二部分32中的哪些位表示R、S及V中的每一者。

[0156] 例如，若control3域的位<12>及位<5>具有零值，则这指示数据元素大小N为32位。因此，指令译码器6或处理电路4亦可确定：旋转参数R具有control3域的位<10:6>的值，大小参数具有control3域的位<4:0>的值，且反转信息V具有二进制值0bv00000，其中v为control3域的位<11>的值。类似地，对其他数据元素大小而言，可以按图示于图5中的方式，从control3域来确定参数N、参数R、参数S及参数V。

[0157] control3域包括子部分(位<11:6>)，该子部分表示参数R及V。数据元素大小N愈大，则旋转参数R所需的位数目愈大而反转信息V所需的位数目愈小。因此，按图示于图5中的方式，在这些参数之间共享control3域的位<11:6>。对不同的数据元素大小N而言，将子部分的位分配为指示旋转参数R或指示反转信息V。以类似方式，指示最高有效位位置S的第二部分32的部分及第一部分30可共享固定数目的位，因为用于这些值中的每一个值的位数目随着数据元素大小N增大或减小而沿相反方向变化。由于对于任何给定数据元素大小，一个参数不需要的位被用以指示另一参数，则未剩下未指示数据元素大小的一些值的任何参数的位，且因此control3域的编码高效地表示参数N、参数R、参数S及参数V。

[0158] 下文将结合图10更详细地描述对control3域的说明。

[0159] 图6图示出根据位域位置参数R及最高有效位位置S执行的位域操纵操作的示例。源位域开始位置A、结果位域开始位置B及位域宽度W都可从参数S和R得出。

[0160] 位域操纵操作具有视参数S与参数R间的关系而定的两种替代形式。响应于该指令的位域提取形式BFX，控制处理电路4以提取位于第一源数据元素src1内的选定位置处的位域bf且将该位域写入相应结果数据元素的最低有效部分。响应于该指令的位域插入形式BFI，控制处理电路4以复制位于第一源数据元素src1的最低有效部分的位域且将经复制的位域插入相应结果数据元素内的选定位置。

[0161] 如图6的上面部分所示，若S≥R，则位域操纵指令为位域提取形式BFX。对位域提取形式的指令而言，旋转参数R指示第一源数据元素src1内的位域bf的最低有效位的位置，且最高有效位位置S指示第一源数据元素src1内的位域bf的最高有效位的位置。因此，源位域开始位置A等于旋转参数R，结果位域开始位置B为0，且位域宽度W＝S+1-R。此举意味着：响应于位域提取形式BFX的指令，将结果数据元素的位<S-R:0>设定为等于相应第一源数据元素src1的位<S:R>。

[0162] 另一方面，若S＜R，则位域操纵指令采用位域插入形式BFI，如图6的下面部分所示。在此情况下，则源位域开始位置A等于0，结果位域开始位置B＝N-R，且位域宽度W＝S+1。因此，在位域插入形式的指令中，将结果数据元素的位<N-R+S:N-R>设定为等于第一源数据元素src1的位<S:0>。

[0163] 通过将源开始位位置A及结果开始位位置B中的至少一个限制为等于0，则图5的编码变得高效，因为仅仅两个参数R、S可对三个参数A、B、W编码以识别待执行的位域操纵。若希望更一般的位域操纵操作，在该位域操纵操作中将来自非零的源开始位位置A的位域写入非零的结果位位置B(如图4中的示例)，则其可通过执行位域提取形式BFX的指令随后执行位域插入形式BFI的指令来实现。

[0164] 图6图示在位域提取形式BFX的位域操纵指令中，将位域bf写入结果数据元素的最低有效部分，且因此结果数据元素可包括前缀部分而非后缀部分。对R及S的一些值而言，亦可能不存在前缀部分。相反地，对位域插入形式BFI的指令而言，可将位域插入于结果数据

元素的任何部分，且因此结果数据值可具有前缀部分及后缀部分两者(R及S的值将控制结果数据元素是包括前缀部分及后缀部分中的零个、一个还是两个)。通常，若B+W＜N，则结果数据元素将包括前缀部分(位<N-1:B+W>)，且若B＞0则将包括后缀部分(位<B-1:0>)。如上所述，位域操纵指令包括子作业码域opc。该opc域控制处理电路4产生选定形式的前缀值或后缀值。图7图示出可针对opc域的不同值而被选择的不同种类的前缀值及后缀值。

[0165] 当子作业码域opc具有0b00的值时，则前缀部分p为位域bf的符号扩展，且后缀部分s的位具有零值。符号扩展位具有这样的值，所述值确保在位域bf插入结果数据元素内时保留位域bf的符号。通常，前缀部分的符号扩展位各自等于位域bf的最高有效位的值(亦即，各自等于第一源数据元素的位<S>)。然而，前缀部分p内的符号扩展位的确切性质将视用于位域bf的特定符号表示法而定。通过将子作业码域设定为0b00，位域操纵指令可用于从第一源数据元素提取包含带符号数据值的位域，且将该值复制至结果数据元素的一部分，同时保留位域的符号。

[0166] 当子作业码域具有0b01的值时，则结果数据元素的前缀部分及后缀部分采用第二源数据元素src2的相应位的值。因此，该形式的指令具有以下效应：将来自第一源数据元素src1的位域bf插入第二源数据元素src2内的位置中，同时使第二源数据元素src2的其他位无变化。在图2中所图示出的示例中，第二源数据元素src2为在执行位域操纵指令之前储存于目的地寄存器Rd中的值，且因此对opc＝01而言，该指令将第一源数据元素src1的位域有效地插入目的地寄存器Rd的给定位置中，同时使其他位无变化。

[0167] 若子作业码域具有0b10的值，则前缀部分及后缀部分的位各自具有零值。因此，对此种指令而言，由结果数据元素中的零来隔离且包围第一源数据元素src1的位域。

[0168] 在图7的示例中，子作业码域opc的0b11值为未定义的，但应了解，其他功能可被与0b11值相关联。

[0169] 如上所述，control3域指示用于控制数据元素在结果数据值内的排序的反转信息V。图8A示意性图示出根据反转信息V将数据元素重新排序的示例。结果数据值等效于在中间结果数据值经受了一系列重新排序迭代的情况下将产生的值，该中间结果数据值是使用如上所述的位域操纵产生的且包括具有与第一源数据值及第二源数据值内的相应数据元素的次序相对应的次序的数据元素，所述一系列重新排序迭代是根据反转信息V来控制的。每一个重新排序迭代与指示粒度的特定组大小相关联且亦与反转信息V的相应位相关联，该粒度用于在数据值内将位组重新排序。每一个重新排序迭代包含以下步骤：确定反转信息V的相应位是否具有预定值，且若相应位具有预定值，则在结果数据值内将相应组大小的相邻对的位组的位置相交换。在如下所述的示例中，可将“1”值用作反转信息V的预定值，但亦可改为使用“0”值。

[0170] 图8A图示出可应用于中间结果以产生结果数据值的一系列重新排序迭代的示例。在重新排序迭代0中，若反转信息V<5:0>的位V<0>具有1值，则将中间结果的数对相邻位的位置交换。亦即，中间结果的位<0>与位<1>交换，且类似地，位<3>与位<2>交换，位<5>与位<4>交换等。另一方面，若位V<0>具有0值，则不执行交换。

[0171] 以类似方式，对图8A中所图示的重新排序迭代1而言，若反转信息V的相应位V<1>具有1值，则将前一迭代结果内的相邻2个位的组的位置交换。因此，位<1:0>与位<3:2>交换，位<7:6>与位<5:4>交换等。相反，若V<1>＝0，则前一迭代结果的位保持不变。

[0172] 类似地，对每一个连续的重新排序迭代而言，若反转信息V的相应位具有1值，则将相邻对的位组的位置交换，其中每一个重新排序迭代的组大小呈2的幂数增大。可将图8A中所图示的运算概括(例如)为一系列重新排序迭代，其中若反转信息的相应位V<i>具有1值，则第i个重新排序迭代将数对的2i位的相邻组的位置交换，而若相应位V<i>具有0值，则不交换数对的2i位的组的位置。尽管使用反转信息V的位<i>来指示第i个重新排序迭代是否应交换位组的位置较方便，但亦可使用反转信息V的位与重新排序迭代间的不同的对应性。重新排序迭代的总数视存在多少数据元素而定。例如，在32位数据值的情况下，将不会执行图8A中所图示的迭代5，因为将仅存在单个32位的组，且因此将不可能交换一对32位组。

[0173] 图8A图示出了可将具有位数目1、2、4等的位组的位置交换的示例。在一些示例中，可对位域操纵指令编码，以使仅与数据元素大小的倍数相对应的位组可进行位置交换。在图8B中图示出了此种示例。

[0174] 图8B图示出64位数据值包含八个8位数据元素A至H的示例。在图5中所图示出的示例中，对control3域编码，以便将零值置放于与用于交换小于数据元素大小N的位组的重新排序迭代相对应的反转信息V<5:0>的任何位中。因此，control3域仅指示反转信息V的位值，所述值指示等于或大于数据元素大小N的位组的交换。例如，对如图8B中所图示出的8位的数据元素大小而言，则将反转信息的位V<2:0>设定为0，以指示不应执行如图8A中所示用于交换1、2及4位的组的重新排序迭代0、1及2。将反转信息V的位<5:3>编码于control3域内，以指示是否应执行重新排序迭代3、4及5来交换8、16及32位的相邻组的位置。

[0175] 图8B图示出第一源数据值src1及第二源数据值src2最初如何包括数据元素A至H，所述数据元素A至H具有如由字母A至H所指示的特定储存次序。若不执行数据元素重新排序，则将位域操纵指令应用于源数据值将产生结果数据值，在该结果数据值中按与源数据值内的数据元素的次序相对应的次序A、B、C、……、H将结果数据元素排序(参见图示出于图8B中的中间结果)。

[0176] 然而，在反转信息V的控制下，当产生最终结果数据值时，指令译码器6可控制处理电路4应用数据元素重新排序。由于反转信息V的位<2:0>具有0值，所以可执行的第一重新排序迭代与位V<3>相关联。图8B图示：若位V<3>具有1值，则将相邻8位组(亦即，相邻数据元素)的位置交换，而同时中间值无变化。随后，若位V<4>具有1值，则另一个重新排序迭代将相邻16位组(亦即，两个数据元素的相邻组)的位置交换。若位V<5>具有1值，则又一个重新排序迭代将相邻32位组(亦即，四个数据元素的相邻组)交换。通过根据反转信息V的位选择性地交换或不交换经不同大小调整的位组，可将不同次序的数据元素提供于结果数据值内。例如，位于图8B的底部的表展示了不同的数据元素排序，所述数据元素排序针对重新排序信息V<5:3>的不同值而产生于图示于图8B中的重新排序迭代。因此，可见，通过应用连续的重新排序迭代，每一个迭代将相邻对的经不同大小调整的位组的位置交换，随后可在结果值内产生一系列数据元素排序。

[0177] 应注意，对数据处理电路4而言，实际上不必使用位域操纵操作产生中间结果并且随后将该一系列重新排序迭代应用于中间结果以产生结果数据值。数据处理电路4可被配置为在对数据元素重新排序的同时应用位域操纵，以使得产生源数据元素的位域被包括于每一个结果数据元素内的结果数据值，且以与反转信息V相对应的次序将结果数据元素排序，而从不产生任何中间结果。结果数据值可能仅为这样的值，该值等效于在将重新排序迭

代应用于此种中间结果的情况下会获得的结果。

[0178] 图9图示出处理如图2的示例中那样被编码的位域操纵指令的示例。应注意，尽管图9图示出了展示一系列方法步骤的流程图，但实际上，用于实施指令的硬件可彼此并行执行这些步骤中的若干个步骤或该硬件可执行产生类似结果的其他步骤(对图示于本申请案中的其他流程图而言同样如此)。

[0179] 在步骤50中，指令译码器6检查子作业码域opc的值。若子作业码域opc具有0b00的值，则方法行进至步骤52，在步骤52中将结果数据值初始化为零值。随后，在步骤54中，将布尔量延伸(Boolean quantity extend)设定为TRUE(真)值(指示当应用位域操纵时前缀值被设定为经提取的位域的符号扩展)。

[0180] 另一方面，若在步骤50中，将子作业码域opc确定为具有0b01值，则在步骤56中，将结果数据值初始化为第二源数据值src2的值(以使每一个结果数据元素将具有这样的前缀部分及/或后缀部分，所述前缀部分及/或该后缀部分具有与第二源数据值src2的相应数据元素的那些位值相对应的位值)。随后，在步骤58中，将布尔量延伸设定为FALSE(假)值，以指示当产生前缀部分时将不执行符号扩展。

[0181] 或者，若在步骤50中，发现子作业码域opc具有0b10值，则在步骤60中将结果值设定为零值，且在步骤62中将布尔值延伸设定为FALSE值。因此，当产生每一个结果数据元素时，除插入的位域以外的任何位将具有零值，且将不执行位域的符号扩展。

[0182] 不管在步骤50中子作业码域opc的值为如何，随后方法行进至步骤64，在步骤64中执行函数TRIDECODE(control13)以将位域操纵指令的控制域control13译码，以识别旋转参数R、最高有效位位置S、反转信息V及数据元素大小N。下文将就图10来描述TRIDECODE函数。

[0183] 在步骤64中识别R、S、V及N的值后，随后在步骤66中，确定最高有效位位置S是否大于或等于旋转参数R。若S大于或等于R，则在步骤68中将源位字段置值A设定为等于R，将结果位域位置值B设定为0且将位域宽度W设定为等于S+1-R(亦即，位域操纵为位域提取形式BFX)。另一方面，若在步骤66中确定S小于R，则指令具有位域插入形式BFI，且因此在步骤70中，将源位域位置值A设定为0，将结果位域位置值B设定为N-R，且将位域宽度W设定为S+1。

[0184] 在步骤72中，不管指令是具有位域插入形式BFI还是具有位域提取形式BFX，随后处理电路4产生结果数据值，在该结果数据值中每一个数据元素具有位<B+W-1:B>，所述位<B+W-1:B>等于相应第一源数据元素src1的位<A+W-1:A>。此步骤确保：第一源数据元素的目的地位域被复制到结果数据元素内的所希望位置。结果数据元素的剩余位继续具有如在步骤52、步骤56及步骤60中的一个步骤中所初始化的值。

[0185] 随后在步骤74中，确定布尔量延伸是否为TRUE且B+W＜N。若延伸为TRUE且B+W＜N，则此举指示：在结果数据元素内存在前缀部分<N-1:B+W>且子作业码域opc指示了符号扩展已被应用于位域。在此情况下，在步骤76中，处理电路4将每一个结果数据元素的位<N-1:B+W>设定为从源数据元素src1复制来的位域内的符号位的值(亦即，将第一源数据元素的位<S>复制于结果数据元素内的前缀部分的每一个位处)。此举具有以下效应：保留从第一源数据元素src1提取的位域的符号。另一方面，若在步骤74中，布尔量延伸为FALSE，则省略步骤76，且因此结果数据元素的前缀部分将保留其先前值(如在步骤60中所设定的零值或如在步骤56中所设定的第二源数据元素src2的先前值)。若B+W＝N，则亦省略步骤76，因为在此

情况下将不存在前缀部分。

[0186] 在步骤78中，例如如上文结合图8A及图8B所描述的，基于从位域操纵指令的control3域所识别的反转信息V，来应用结果数据值内的结果数据元素的重新排序。应注意，在一些硬件实施方案中，可与步骤72中产生结果数据元素并行执行此步骤。

[0187] 因此，在位域操纵指令的控制下，产生这样的结果数据值，其中，每一个结果数据元素包括取自相应第一源数据元素src1内的所希望位置<A+W-1:A>的位域；其中，将位域限定于结果数据元素的内的前缀部分及后缀部分被设定为零值、符号扩展值或第二源数据元素src2的值；并且其中，可选地，还执行该结果内的结果数据元素的重新排序。因此，位域操纵指令提供共同编码内的大范围的位域操纵。

[0188] 此外，通过将参数S及参数R设定为合适的值以使位域对应于整个第一源数据元素src1，位域操纵指令亦可用作数据元素重新排序指令。在此情况下，结果数据元素与相应第一源数据元素相同，但是根据反转信息V在结果数据值内被重新排序。

[0189] 图10图示出如图9的步骤64所示(并且，如将在下文图15的步骤160中所述)的使用TRIDECODE函数将control3域译码的示例。可通过指令译码器6、处理电路4、或指令译码器6与处理电路4的组合，执行control3域的译码。如下文将描述的，control3域可用于不同种类的指令。结合图10所提及的“数据大小”对应于参看图2至图9所描述的位域操纵指令的数据元素大小N，且对应于参看图11至图15所描述的按位逻辑指令的模式重复大小N。

[0190] 在图10的步骤100中，将参数V<5:0>初始化为0b000000的全零值。此举确保：未被编码于control3域内的反转信息V的任何值将采用零值，且因此将确保：将不会执行相应重新排序迭代。

[0191] 在步骤102中，通过将control3域的位<12>与control3域的位<5:0>的反序的相串接，来确定临时值temp<6:0>，且将长度参数len确定为temp<6:0>内具有“1”值的最高(最高有效)位的位位置。temp<6:0>值表示control3域中的第一部分30可驻留于的一部分的位值。若数据大小为64位，则具有“1”值的最高有效位将为temp<6:0>的位<6>，且因此len＝6。对其他数据大小而言，temp<6:0>内的最高的“1”位将对应于control3域的位<5:0>内的最高的“0”的位置，且因此对N＝32、16、8、4、2、1而言len分别为5、4、3、2、1、0(参见图5及图12)。因此，长度参数len为第一部分30的大小的指示，其允许在图10的步骤104至步骤114中确定参数R、参数S及参数V的编码格式以及数据大小N。将了解，在control3域的译码期间产生临时值temp并非必要的，且在一些实施例中，单个操作可从control3域的值直接确定参数len。

[0192] 在图10的步骤104中，通过将0b1值向左移动len位位置，来确定数据大小N。例如，若len＝2，则将0b1值向左移动2会产生0b100值，亦即，四个位的数据大小N。因此，数据大小N等于2len。以此方式，control3域的第一部分30的大小可与如图5及图12中所示的相应数据大小N相关。

[0193] 在图10的步骤106中，确定长度参数len是否等于0。若len＝0，则在步骤108中，将旋转参数R及最高有效位位置S设定为0。此举对应于1位的数据元素大小，因此仅存在可被应用的一个可能的位域操纵：将每一个第一源数据元素src1的单个位复制至相应结果数据元素src2的单个位，且因此对N＝1而言，不需要任何旋转参数R及最高有效位位置参数S(应注意，在此情况下control3域的编码使用用于指示R、S的位来改为指示反转信息V及第一

部分30)。

[0194] 若在步骤106中长度参数len不等于零,则在步骤110中,将旋转参数R设定为具有control3域的位<len+5:6>的值的无符号整数,同时将最高有效位位置S设定为具有control3域的位<len-1:0>的值的无符号整数。此举对应于如图5及图12中所示的control3域的编码。

[0195] 在步骤112中,确定长度参数是否小于6。若如此,则在步骤114中,将反转信息V的位<5:len>设定为等于control3域的位<11:len+6>。反转信息的位<len-1:0>保留如在图10的步骤100中所初始化的其零值。再次地,此举对应于如图5中所示的control3域的编码。

[0196] 另一方面,若在步骤112中len=6,则数据大小为与整个64位数据值的大小相对应的64位,对图5的示例而言64位为最大数据元素大小。若len=6,则不可能存在数据元素的任何重新排序,因为仅存在数据值内的单个数据元素的一种排序方式。因此,若len=6,则省略图10的步骤114,从而使重新排序信息V继续等于如在步骤100中所设定的0。

[0197] 最后,在步骤116中,将参数R、参数S、参数V及参数N的所确定值返回,以供处理电路4在处理含有control3域的指令时使用。

[0198] 如图5中所示的control3域的编码可用于其他类型的指令以及位域操纵指令。control3域的编码适用于如下的任何种类的指令,该种指令指定一组数据大小中的选定的一个数据大小及至少一个控制参数,该至少一个控制参数的位数目根据选定数据大小而变化。位域操纵指令为SIMD指令的示例。类似地,control3编码可用于另外的指令,该指令指定数据元素大小及至少一个其他参数,该至少一个其他参数的位数目根据数据元素大小而变化。

[0199] 图11图示出按位逻辑指令LOGIC,该按位逻辑指令LOGIC为使用control3编码的指令的另一个示例。将按位逻辑指令的control3域如图12中所示那样进行编码。图5与图12的比较显示:对按位逻辑指令LOGIC而言以与对位域操纵指令BF而言相同的方式识别参数R及参数S,并且对按位逻辑指令而言以与图5中对位域操纵指令而言识别数据元素大小N的方式相同的方式识别模式重复大小N。按位逻辑指令不等效于反转信息V,且因此在按位逻辑指令LOGIC的control3域的编码中,不使用control3域的表示位域操纵指令BF的反转信息V的位。将了解,在其他实施例中,在图12中图示为x的未使用的位可指示另外的参数。

[0200] 图13及图14图示出按位逻辑指令LOGIC的功能。指令识别源寄存器Rn,该源寄存器Rn储存第一源数据值src1。按位逻辑指令的子作业码域opc指示待应用于第一源数据值src1及掩码数据值的逻辑运算的类型。例如,逻辑运算可为与运算、或运算及异或(XOR)运算中的一种运算,如图14中所示。

[0201] 指令译码器6通过产生控制信号来对逻辑指令作出响应,所述控制信号用于控制处理电路4通过使用选定逻辑运算将源数据值src1与掩码数据值(掩码)相组合来产生结果值。掩码数据值是使用如从逻辑指令的control3域识别的选定数据大小N以及参数R及参数S确定的值。

[0202] 掩码数据值包含重复模式的位。重复模式的重复单位为通过control3参数识别的选定数据大小N。每一个重复模式由N个位组成,在所述N个位中,S+1个位具有“1”值而其

他位具有“0”值。重复模式等效于在(S+1)个位位于该模式(其中较高有效位具有“0”值)的最低有效端并且随后该模式被向右旋转R个位位置的情况下将得到的值,所述(S+1)个位各自具有1值。将了解,在其他实施例中,旋转参数R可指示左旋转而非右旋转的量。如图13中所示,使重复模式贯穿掩码数据值重复M/N次(其中M为由src1及掩码数据值所包含的位数目)。因此,被编码于control3域内的N值、R值及S值使得能产生各种不同的掩码,以供使用逻辑运算将所述掩码与源数据值src1相组合。

[0203] 如图14中所示,使用按位与运算、或运算或异或(XOR)运算中的一种运算将掩码值与源数据值src1组合。逻辑运算可用于(例如)测试、设定、清除或反转数据值的特定部分的位值,或可用于隔离数据值的特定部分。应注意,实际上不必通过处理电路4产生掩码数据值,而是,处理电路4可能仅产生具有某种形式的最终结果,该种形式的最终结果等效于使用逻辑运算将掩码与源数据值src1相组合的结果。

[0204] 图15图示出处理图示于图11至图14中的形式的按位逻辑指令的方法。在步骤150中,确定子作业码域opc的值。若子作业码域具有0b00的值,则在步骤152中将逻辑运算确定为与运算。若子作业码域opc具有0b01的值,则在步骤154中将逻辑运算确定为或运算。若子作业码域opc具有0b10的值,则如在步骤156中所确定的,逻辑运算为异或(XOR)运算。在该示例中保留0b11的opc值,但在另一个实施例中,可将0b11的opc值指派至不同形式的逻辑运算。

[0205] 不管子作业码域opc的值为如何,随后在步骤160中,使用如结合图10所描述的TRIDECODE函数来从control3域确定参数R、参数S、参数V及参数N。在此情况下N表示待应用于源数据值的、掩码内的重复模式的大小。S+1指示掩码值内的“1”位的数目。R指示掩码值内的“1”位的位置。对该逻辑指令而言,用于位域操纵指令的反转信息V并非为所关注的,且因此当计算逻辑运算的结果时可忽略由TRIDECODE函数所返回的V值。

[0206] 在步骤162中,确定位的重复模式。将重复模式初始化为位<N-1:0>,包括具有“0”值的位<N-1:S+1>及具有“1”值的位<S:0>。在步骤164中,将位的重复模式旋转R位个位置(可在需要时向右旋转或向左旋转,但在图13的实施例中是向右旋转的)。

[0207] 在步骤166中,通过贯穿掩码将位的旋转重复模式复制M/N次,来形成掩码数据值,其中M为用于该特定处理操作的数据值大小。例如,可基于由按位逻辑指令的sf域所识别的寄存器存取大小来确定M。随后在步骤168中,结果数据值被生成作为使用在步骤152、154、156之一选定的逻辑运算将第一源数据值src1与掩码数据值相组合的结果。

[0208] 再次,图15的步骤仅为示例且可彼此并行(而非串行)执行。此外,在步骤166中所描述的掩码值实际上可不通过处理电路4产生,而是,在步骤168中处理电路4可直接从源数据值src1及按位逻辑指令的control3域产生结果数据值。

[0209] 因此,不同指令可使用相同格式的control3域,以指示用于控制相关联处理操作的参数。通过(如图5及图12的示例中所示)使用共同编码来指示用于不同种类指令的参数,可共享用于译码及处理这些指令的硬件中的一些硬件,且因此可降低处理电路4及指令译码器6的复杂性。

[0210] 下文指示了用于指示与位域操纵指令及按位逻辑指令相对应的操作的示例伪码。图示于伪码中的操作仅为示例,且对处理装置2的硬件而言不必包括用于实际执行这些步骤的组件。实际上,图示于伪码中的步骤中的一些步骤可彼此并行执行,而非作为一系列连

续步骤执行。然而，伪码将足够由熟习此项技术者用于产生用于产生结果数据值的硬件实施例，该结果数据值等效于伪码中所指示的处理步骤的结果。在伪码中，参数“从”、“到”、“宽度”及“大小”分别对应于如上所述的参数A、参数B、参数W及参数N。参数“数据库大小”指示如上所述的源数据值及结果数据值的大小M。术语“R[n]”及“R[d]”分别表示源寄存器Rn及目的地寄存器Rd。在伪码中，单引号内的一系列1及0(诸如，‘110’)表示二进制记数法的一串连续的位。函数UInt将一串位转换成它们所表示的无符号整数，所以UInt(‘110’)返回值6。

[0211] 伪码的第一部分指示用于对control3域译码的TRIDECODE函数的示例。伪码对应于图10的操作。应注意，TRIDECODE函数随后被用于处理位域操纵指令及按位逻辑指令两者。

```
//对位域直接 control3 译码以提供参数 R、S、V 及 SIZE 控制参数
(integer, integer, bits(6), integer)  TRIDECODE(bits(13) control13)
    integer R;
    integer S;
    integer len;
    bits(6) V=Zeros();
    len=HighestSetBit(control13<12>:NOT(control13<5:0>));
    if len<0 then UNDEFINED;
    if len==0 then
    R=0;
    S=0;
else
     R=UInt(control13<len+5:6>);
     S=UInt(control13<len-1:0>);
  if len<6 then
      V<5:len>=control13<11:len+6>;
   return (R, S, V, 1<<len);
```

[0212]

[0213]

[0214] 伪码中的下一部分对应于位域操纵指令的译码及执行：

```
//位域译码
integer n=UInt(Rn);
integer d=UInt(Rd);
integer datasize=if sf=='1' then 64 else 32;
boolean inzero;
boolean extend;
integer R;
integer S;
bits(6) V;
```

[0215]

```
integer size;
integer from;
integer to;
integer width;

case opc of
    when '00' inzero=true;  extend=true; //如图 7 中的 SBFX/SBFIZ
    when '01' inzero=false; extend=false; //如图 7 中的 BFXIL/BFI
    when '10' inzero=true;  extend=false; //如图 7 中的 UBFX/UBFIZ
    when '11' UNDEFINED;
```

[0216]

```
if datasize==32 then
    //对 32 位数据值而言，不可能存在 64 位数据元素大小，
    //所以 control13 域的位<12>不可能为 1。
if control13<12>=='1' then
     UNDEFINED;
//同样，对 32 位数据值而言，不可能存在 32 位组的任何重新排序，
//所以 control13 域的位<11>亦不可能为 1。
elsif control13<11>=='1' then
     UNDEFINED;

//使用以上定义的 TRIDECODE 函数将 R 值、S 值、V 值及 size 的值
译码
(R, S, V, size)=TRIDECODE(control13);

if S>=R then
    //BFX 情况
    from=R;
    to=0;
    width=(S+1)-R;
else
    //BFI 情况
    from=0;
    to=size-R;
    width=(S+1);

//位域执行
bits(datasize) operand1=R[n];
bits(datasize) result;
integer base;
```

[0217]

```
integer src;
integer dst;
integer vbit;

//若 inzero 为真则将结果初始化为零，或若 inzero 为假则将结果初始化为目的地寄存器的复本
result=if inzero then Zeros() else R[d];

//将来自 src 操作数的每一个数据元素的位域复制至结果的相应数据元素
base=0;
while base<datasize do
    src=base+from;
    dst=base+to;
    result<dst+width-1:dst>=operand1<src+width-1:src>;

//若请求，则执行符号位复制
if extend && to+width<size then
    result<base+size-1:dst+width>=
        Replicate(src<base+S>, size-(to+width));
base=base+size;

//若 V<vbit>=1，则使结果中的相邻对的 2^vbit 位反转
for vbit=0 to 5
     if V<vbit>=='1' then
         bits(datasize) tmp=result;
         size=1<<vbit;
         base=0;
         while base<datasize do
```

[0218]

```
result<base+size-1:base>=tmp<base+(2*size)-1:base+size>;
result<base+(2*size)-1:base+size>=tmp<base+size-1:base>;
base=base+(2*size);

//将结果写入至目的地寄存器
R[d]=result;
```

[0219] 伪码中的下一部分指示用于译码且处理按位逻辑指令的函数：

```
//逻辑（直接）译码

integer R;
integer S;
integer size;
bits(datasize) mask;

integer datasize=if sf=='1' then 64 else 32;
integer n=UInt(Rn);
integer d=UInt(Rd);
```

[0220]

```
LogicalOp opcode;

//opc 定义待应用的逻辑运算的类型
case opc of
    when '00' opcode=LogicalOp_AND;
    when '01' opcode=LogicalOp_OR;
    when '10' opcode=LogicalOp_EOR;
    otherwise UNDEFINED;

//对 32 位数据值而言，不可能存在 64 位模式重复大小
if datasize==32&&control13<12>=='1' then UNDEFINED;
```

[0221]

```
//使用上述 TRIDECODE 函数将 R、S 及 size 译码，且
//忽略 V 的返回值
(R, S,-, size)=TRIDECODE(control13);

//基于 R、S 及 size 确定掩码
bits(size) pattern=Zeros(size-(S+1)):Ones(S+1);
pattern=ROR(pattern, R);
mask=Replicate(pattern, datasize DIV size);

//逻辑（直接）执行
bits(datasize) operand1=R[n];
bits(datasize) operand2=mask;
bits(datasize) result;

//通过使用选定逻辑运算将源操作数与掩码操作数组合，来产生结果
case opcode of
    when LogicalOp_AND result=operand1 AND operand2;
    when LogicalOp_OR resul=operand1 OR operand2;
    when LogicalOp_EOR result=operand1 EOR operand2;

//将结果写入至目的地寄存器
R[d]=result;
```

[0222] 伪码的下一部分指示可使用control13编码的另一种指令的示例。提取指令产生包含若干结果数据元素的数据值，每一个结果数据元素对应于通过将第一源操作数(operand1)的相应第一源数据元素与第二源操作数(operand2)的相应第二源数据元素相串接所形成的值中的选定部分。在该示例中，control13值定义数据元素大小及“lsb”值，该“lsb”值指示串接的第一及第二源数据元素中的哪一部分被包括于结果数据元素中。数据元素大小及“lsb”值对应于从TRIDECODE函数返回的N值及S值，同时忽略由TRIDECODE函数返回的V值及R值。

```
//提取译码
integer datasize=if sf=='1' then 64 else 32;
integer n=UInt(Rn);
integer m=UInt(Rm);
integer d=UInt(Rd);
integer lsb;
integer size;
bits(13) control13;

if datasize==32&&N=='1' then UNDEFINED;

control13=N:Zeros(6):imm6;
```

[0223] (-, lsb,-, size)=TRIDECODE(control13);

```
//提取执行
bits(datasize) operand1=R[n];
bits(datasize) operand2=R[m];
bits(datasize) result;
bits(2*size) concat;
integer bbit=0;

while bbi<datasize
    integer ebit=bbit+size-1;
    concat=operand1<ebit:bbit>: operand2<ebit:bbit>;
    result<ebit:bbit>=concat<lsb+size-1:lsb>;
    bbit=bbit+size;
```

[0224]

```
R[d]=result;
```

[0225] 图16图示可使用的虚拟机实施方式。虽然以上所述的实施例在用于操作支持有关技术的特定处理硬件的装置及方法方面实施了本发明，但亦有可能提供所谓的硬件器件的虚拟机实施方案。所述虚拟机实施方案在主机处理器200上执行，该主机处理器200执行主

机操作系统220，该主机操作系统220支持虚拟机程序240。通常，提供以合理速度执行的虚拟机实施方案需要大功效处理器，但在某些环境中(诸如，当出于兼容性或再使用的原因希望执行由另一个处理器原生的码时)，此种方法可能为合理的。虚拟机程序240提供到应用程序260的应用程序编程接口，该应用程序编程接口与由真实硬件所提供的应用程序编程接口相同，该真实硬件为被虚拟机程序240建模的器件。因此，可使用虚拟机程序240在应用程序260内执行程序指令(包括存储器存取的控制)，以对与虚拟机硬件的互动进行建模。

[0226] 尽管本文已参阅附图详细描写了本发明的说明性实施例，但应理解，本发明并不限于这些精确的实施例，而且在不脱离由所附申请专利范围定义的本发明的范畴及精神的情况下，熟习此项技术者可对本发明进行各种改变及修改。

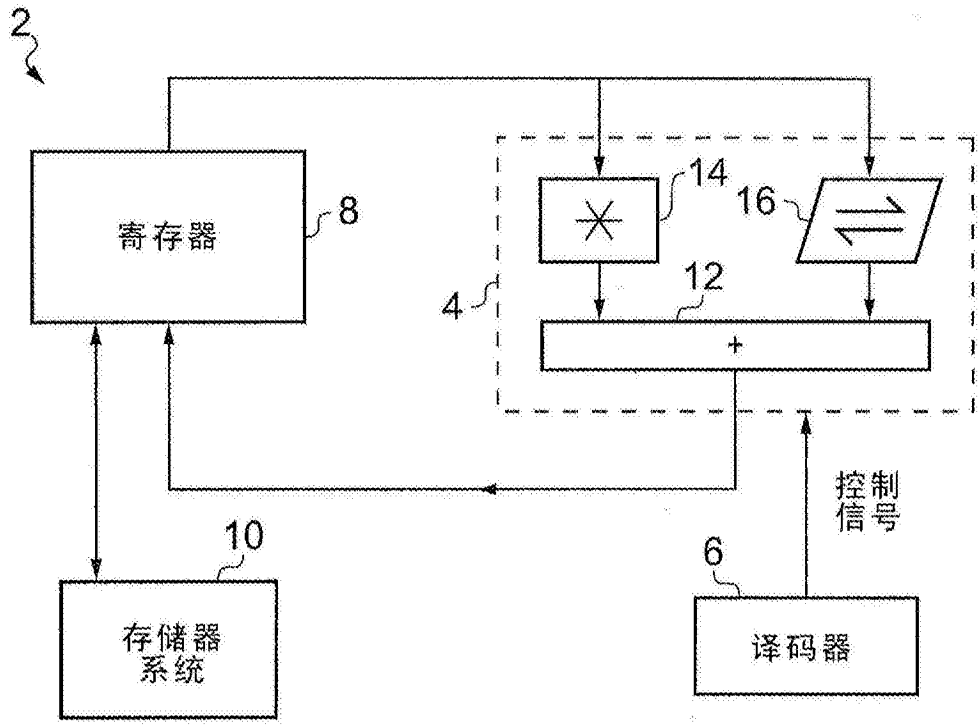

2
寄存器
8
14
16
4
12
+
控制
信号
10
6
存储器
系统
译码器

图1

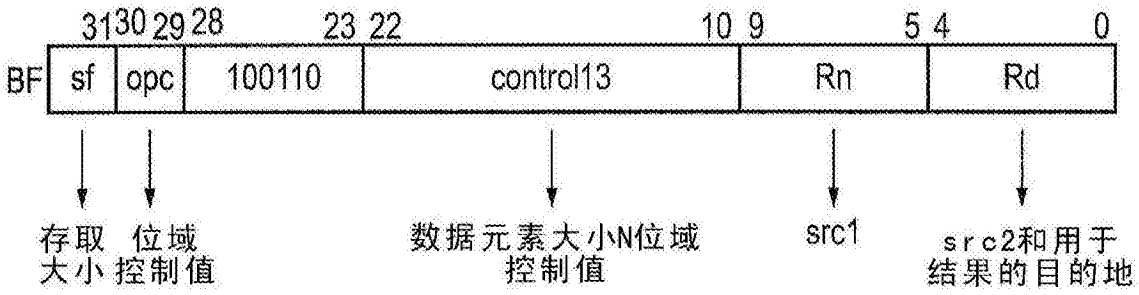

31 30 29 28 23 22 10 9 5 4 0
BF
sf
opc
100110
control13
Rn
Rd
存取
大小
位域
控制值
数据元素大小N位域
控制值
src1
src2和用于
结果的目的地

图2

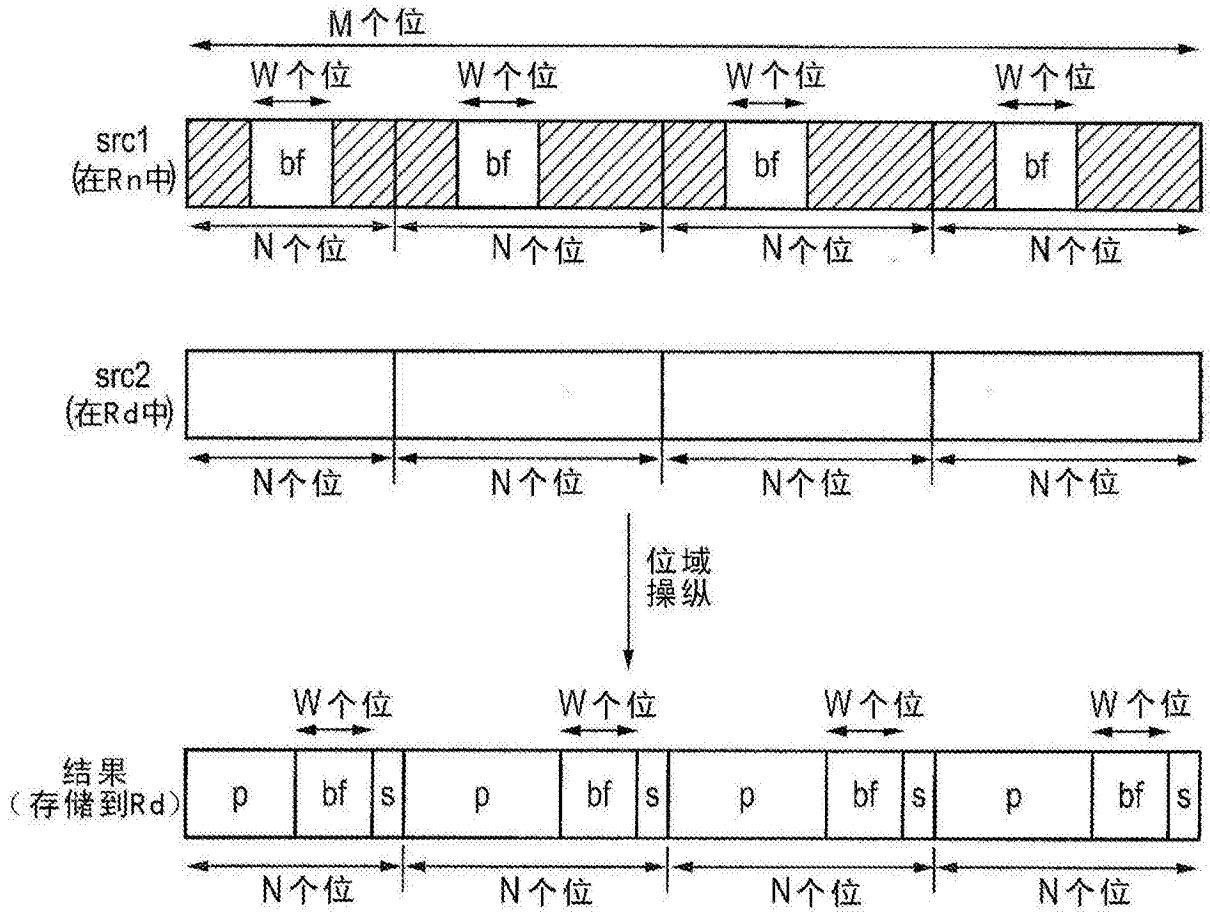

M个位
W个位
W个位
W个位
W个位
src1
(在Rn中)
bf
bf
bf
bf
N个位
N个位
N个位
N个位
src2
(在Rd中)
N个位
N个位
N个位
N个位
位域
操纵
W个位
W个位
W个位
W个位
结果
(存储到Rd)
p
bf
s
p
bf
s
p
bf
s
p
bf
s
N个位
N个位
N个位
N个位

图3

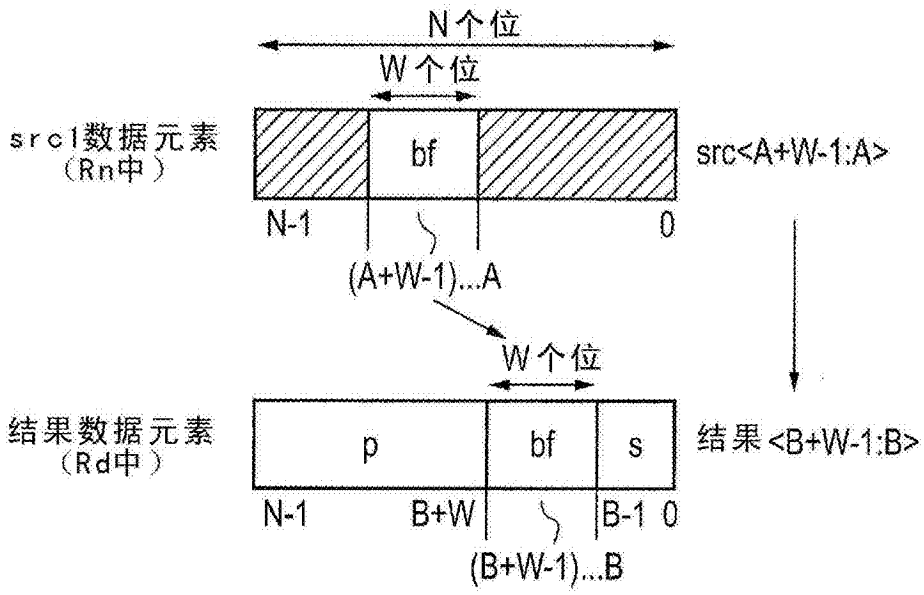

N个位
W个位
src1数据元素
(Rn中)
bf
src<A+W-1:A>
N-1
0
(A+W-1)...A
W个位
结果数据元素
(Rd中)
p
bf
s
结果<B+W-1:B>
N-1
B+W
B-1
0
(B+W-1)...B

图4

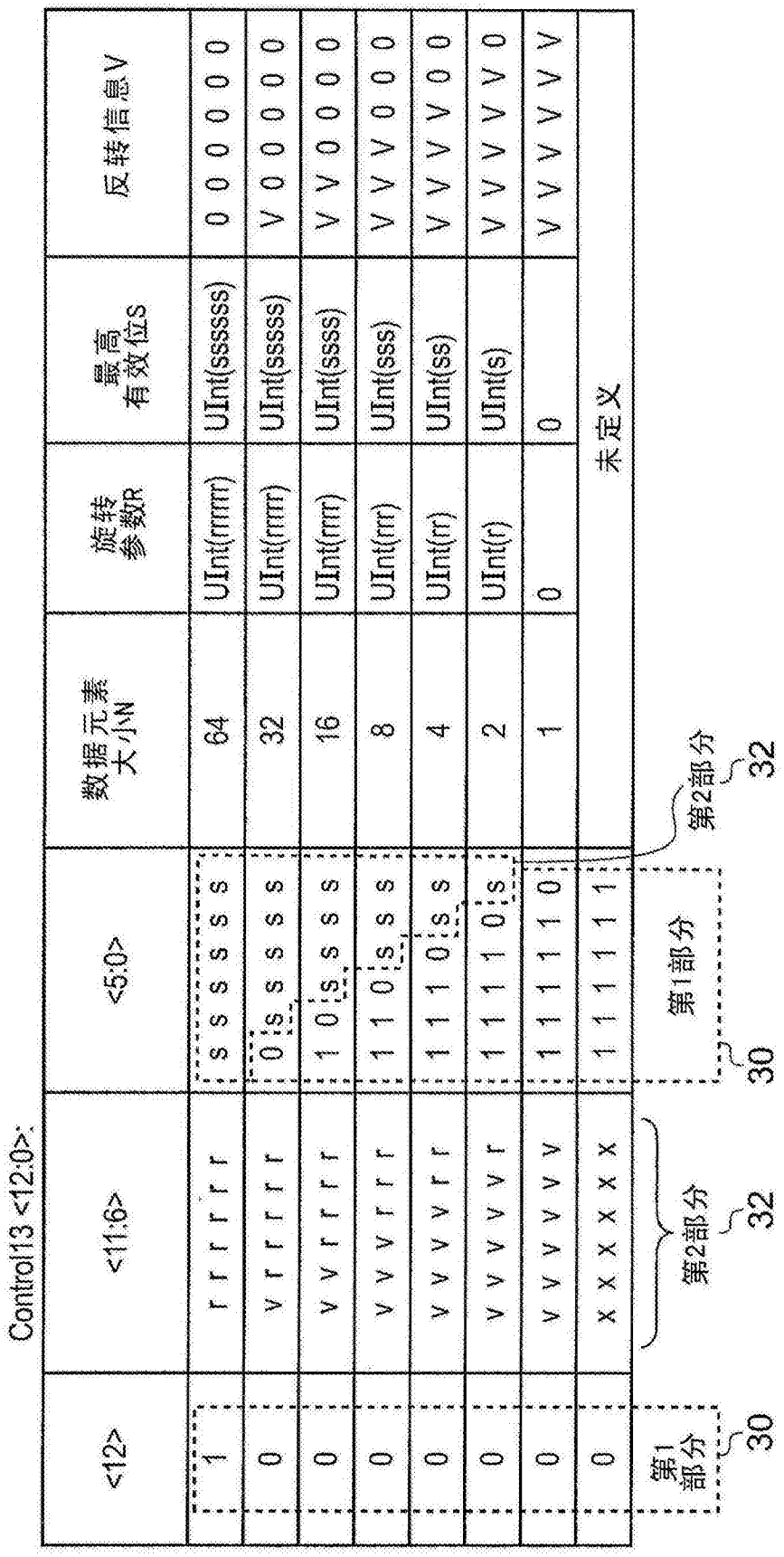
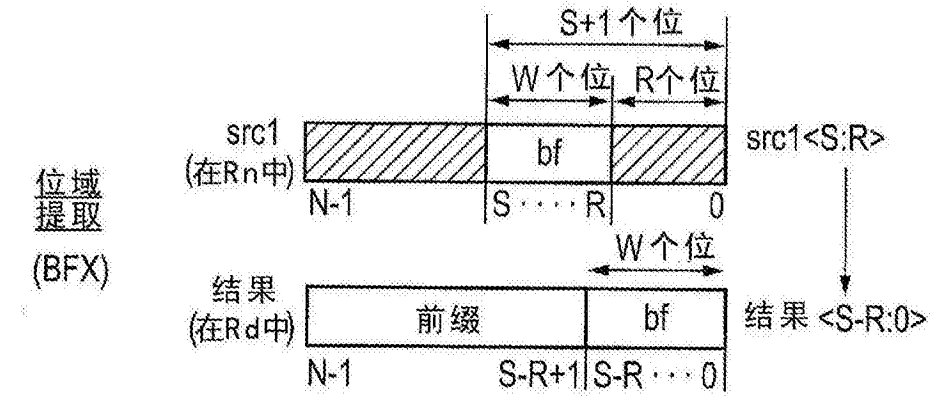

Control13 <12:0>:
<12>
<11:6>
<5:0>
数据元素大小N
旋转参数R
最高有效位S
反转信息V
1 r r r r r r s s s s s s 64 UInt(rrrrrr) UInt(ssssss) 0 0 0 0 0 0
0 v r r r r r 0 s s s s s 32 UInt(rrrrr) UInt(sssss) V 0 0 0 0 0
0 v v r r r r 1 0 s s s s 16 UInt(rrrr) UInt(ssss) V V 0 0 0 0
0 v v v r r r 1 1 0 s s s 8 UInt(rrr) UInt(sss) V V V 0 0 0
0 v v v v r r 1 1 1 0 s s 4 UInt(rr) UInt(ss) V V V V 0 0
0 v v v v v r 1 1 1 1 0 s 2 UInt(r) UInt(s) V V V V V 0
0 v v v v v v 1 1 1 1 1 0 1 0 0 V V V V V V
0 x x x x x x 1 1 1 1 1 1 未定义
第1部分
30
第2部分
32
第1部分
30
第2部分
32

图5

一般位域操纵 ： 结果 <B+W-1:B> = src1<A+W-1:A>

如果 S ≥ R, 则 A = R, B = 0, W = S+1-R



如果 S < R, 则 A = 0, B = N-R, W = S+1

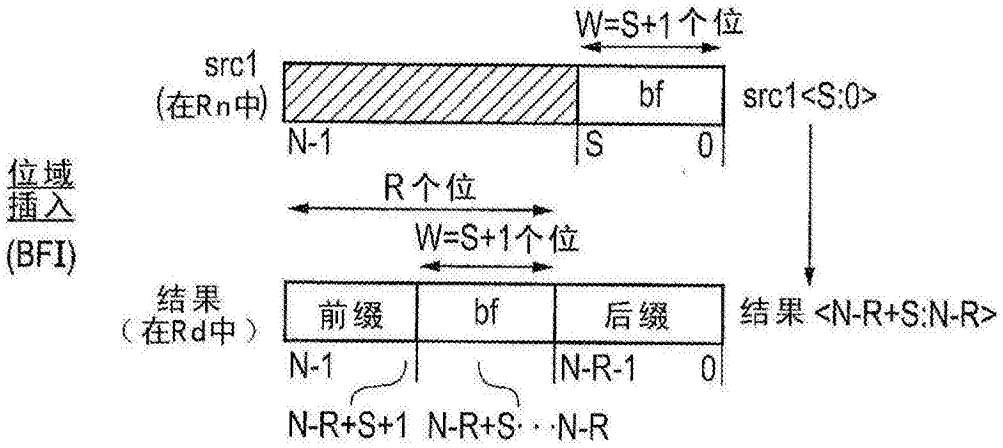


图6

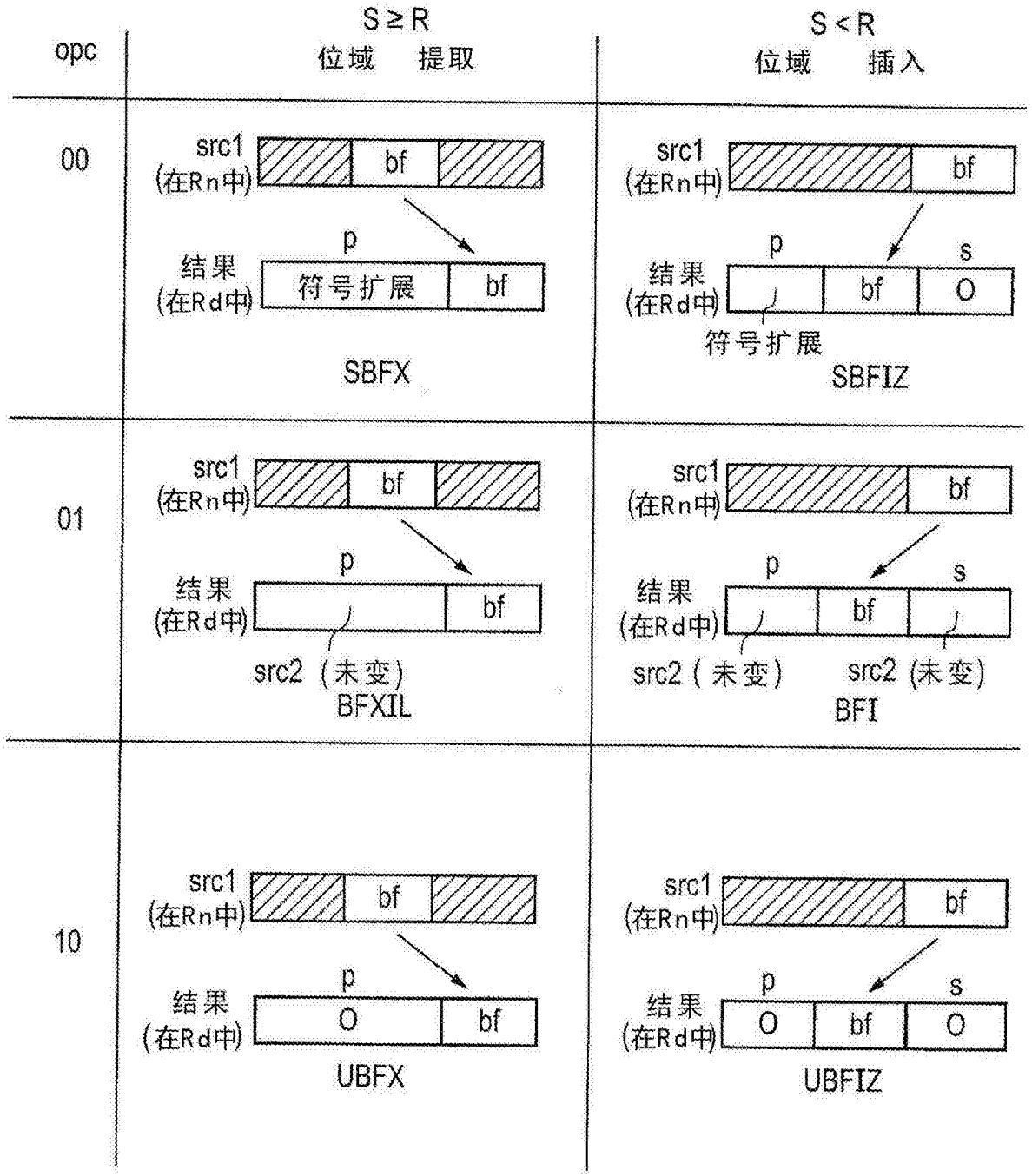

opc
S≥R
位域 提取
S<R
位域 插入
00
src1
(在Rn中)
bf
p
结果
(在Rd中)
符号扩展
bf
SBFX
src1
(在Rn中)
bf
p
s
结果
(在Rd中)
bf
O
符号扩展
SBFIZ
01
src1
(在Rn中)
bf
p
结果
(在Rd中)
bf
src2 (未变)
BFXIL
src1
(在Rn中)
bf
p
s
结果
(在Rd中)
bf
src2 (未变)
src2 (未变)
BFI
10
src1
(在Rn中)
bf
p
结果
(在Rd中)
O
bf
UBFX
src1
(在Rn中)
bf
p
s
结果
(在Rd中)
O
bf
O
UBFIZ

图7

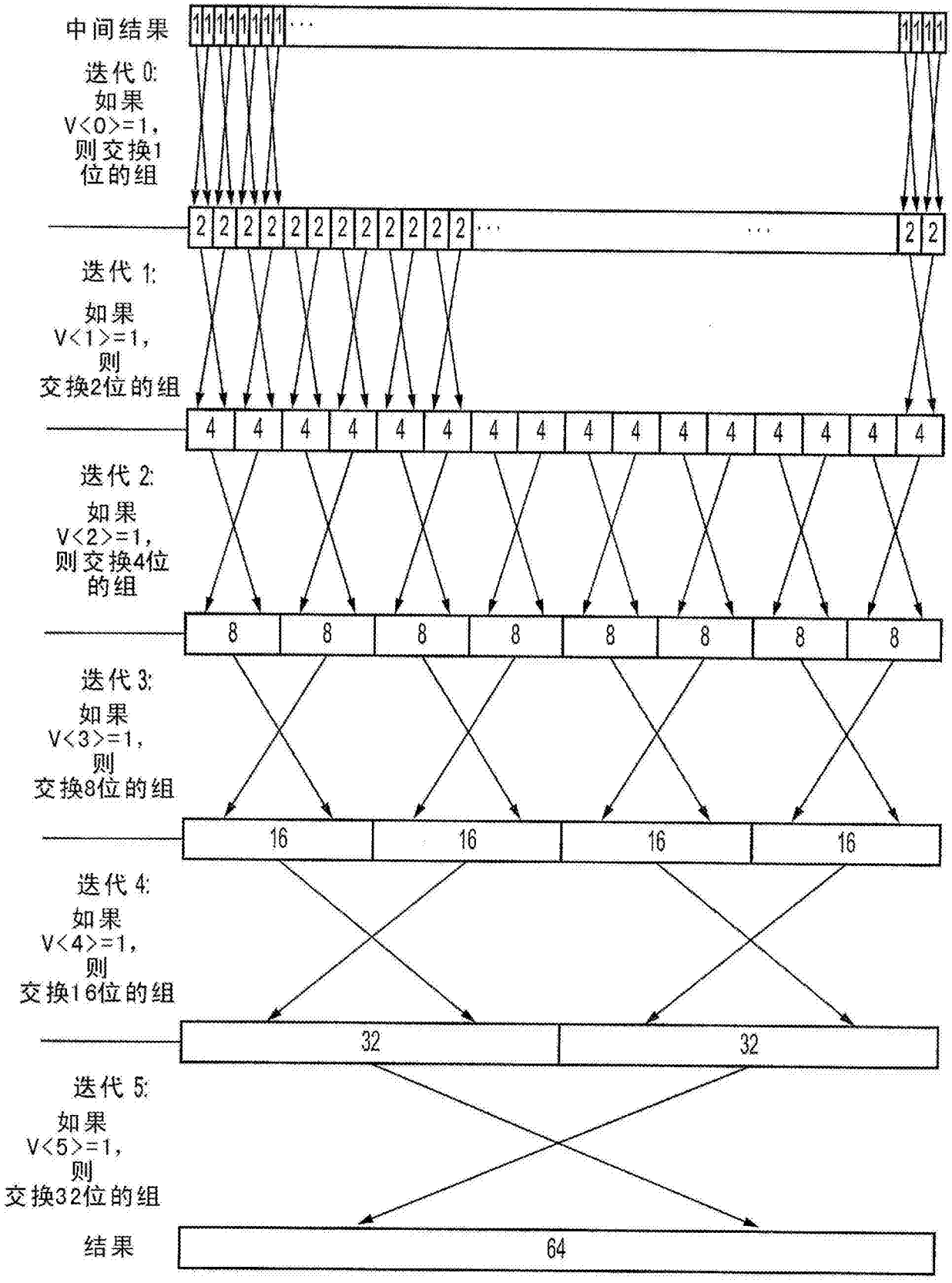
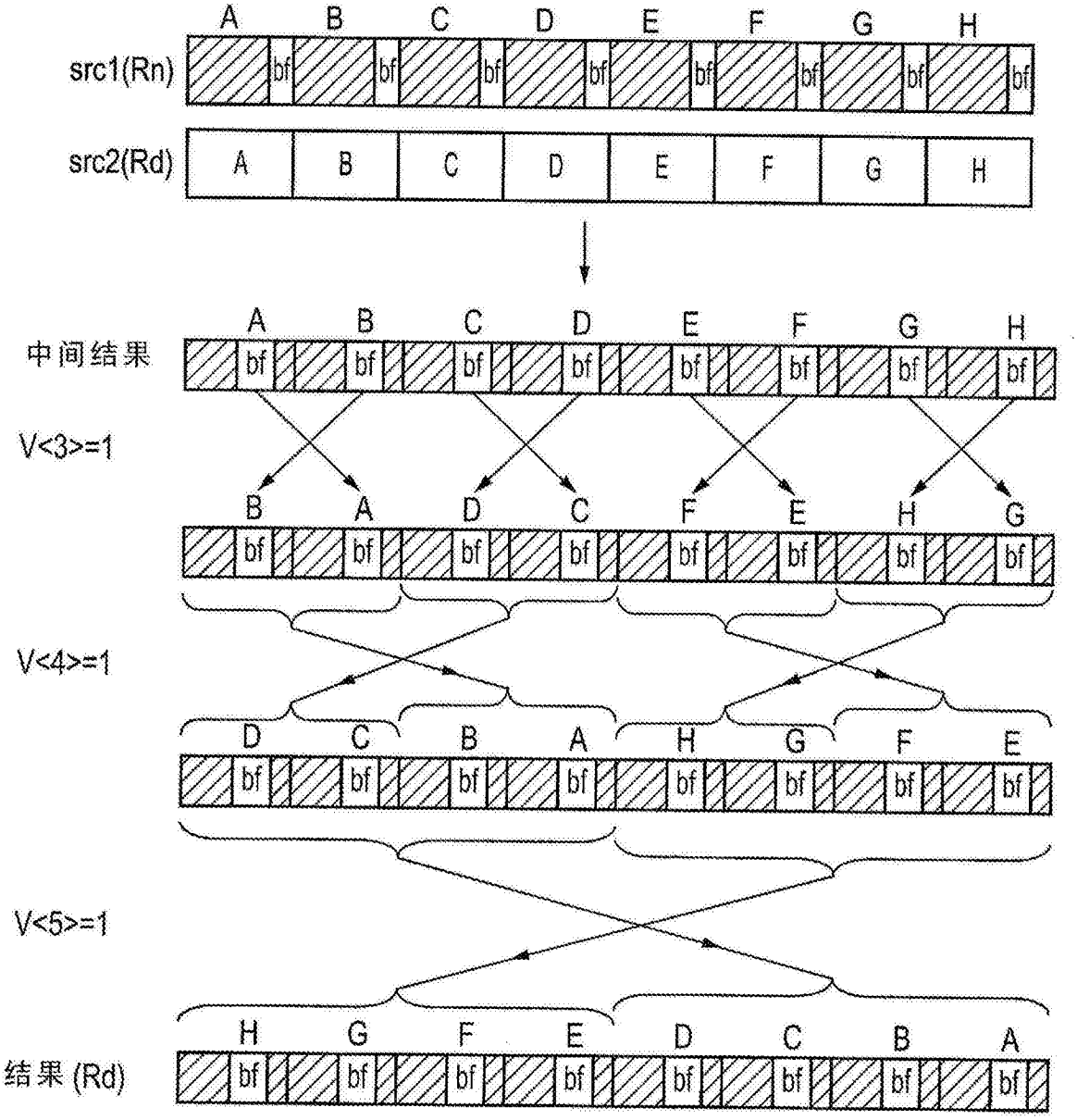

中间结果
迭代 0:
如果
V<0>=1,
则交换1
位的组
迭代 1:
如果
V<1>=1,
则
交换2位的组
迭代 2:
如果
V<2>=1,
则交换4位
的组
迭代 3:
如果
V<3>=1,
则
交换8位的组
迭代 4:
如果
V<4>=1,
则
交换16位的组
迭代 5:
如果
V<5>=1,
则
交换32位的组
结果
64

图8A



V<5:3>	数据元素顺序
0 0 0	A B C D E F G H
0 0 1	B A D C F E H G
0 1 0	C D A B G H E F
0 1 1	D C B A H G F E
1 0 0	E F G H A B C D
1 0 1	F E H G B A D C
1 1 0	G H E F C D A B
1 1 1	H G F E D C B A

图8B

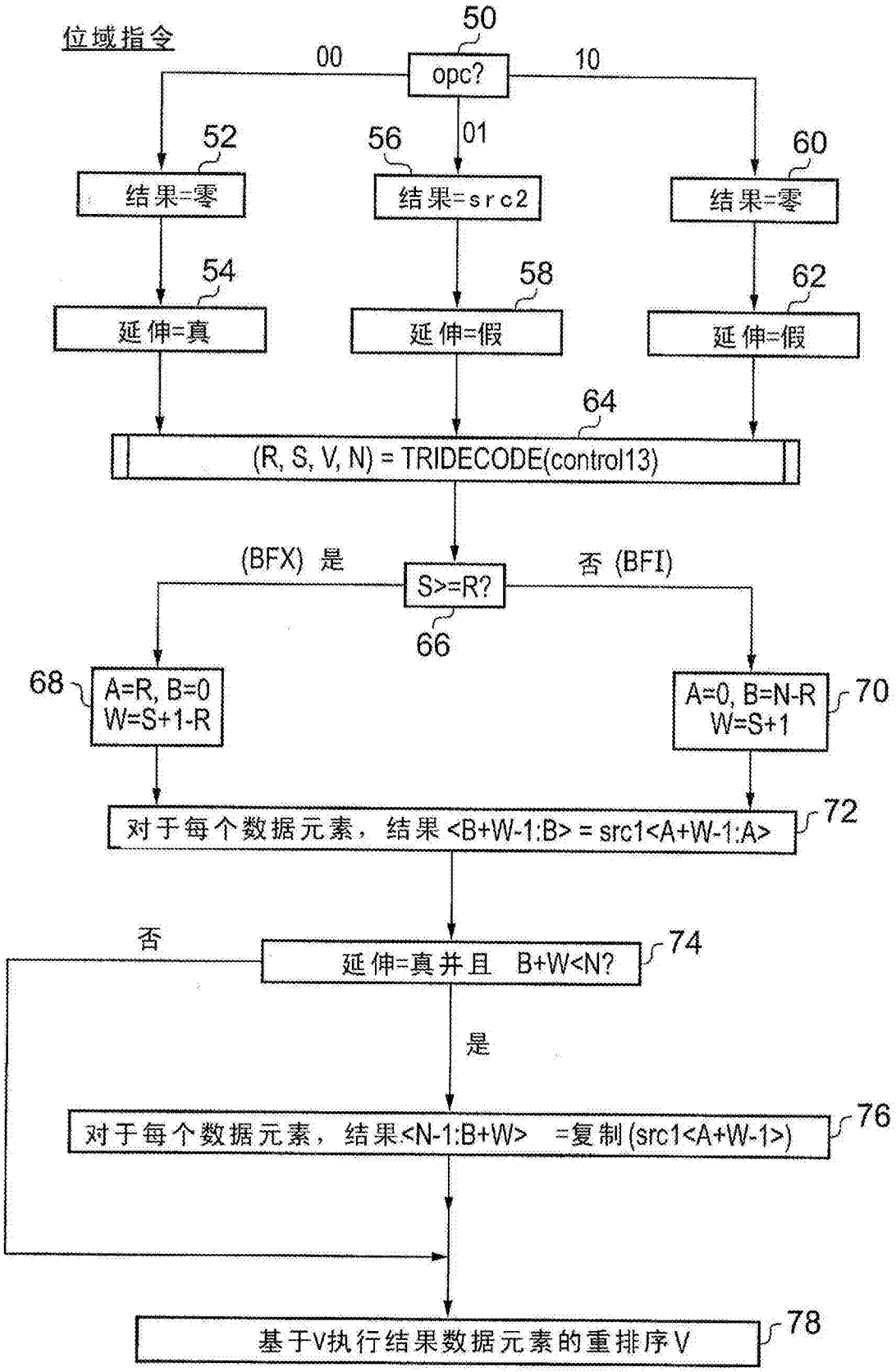

位域指令
50
opc?
00
10
01
52
结果=零
56
结果=src2
60
结果=零
54
延伸=真
58
延伸=假
62
延伸=假
64
(R, S, V, N) = TRIDECODE(control13)
(BFX) 是
S>=R?
否 (BFI)
66
68
A=R, B=0
W=S+1-R
A=0, B=N-R
W=S+1
70
对于每个数据元素，结果 <B+W-1:B> = src1<A+W-1:A>
72
否
延伸=真并且 B+W<N?
74
是
对于每个数据元素，结果<N-1:B+W> =复制(src1<A+W-1>)
76
基于V执行结果数据元素的重排序V
78

图9

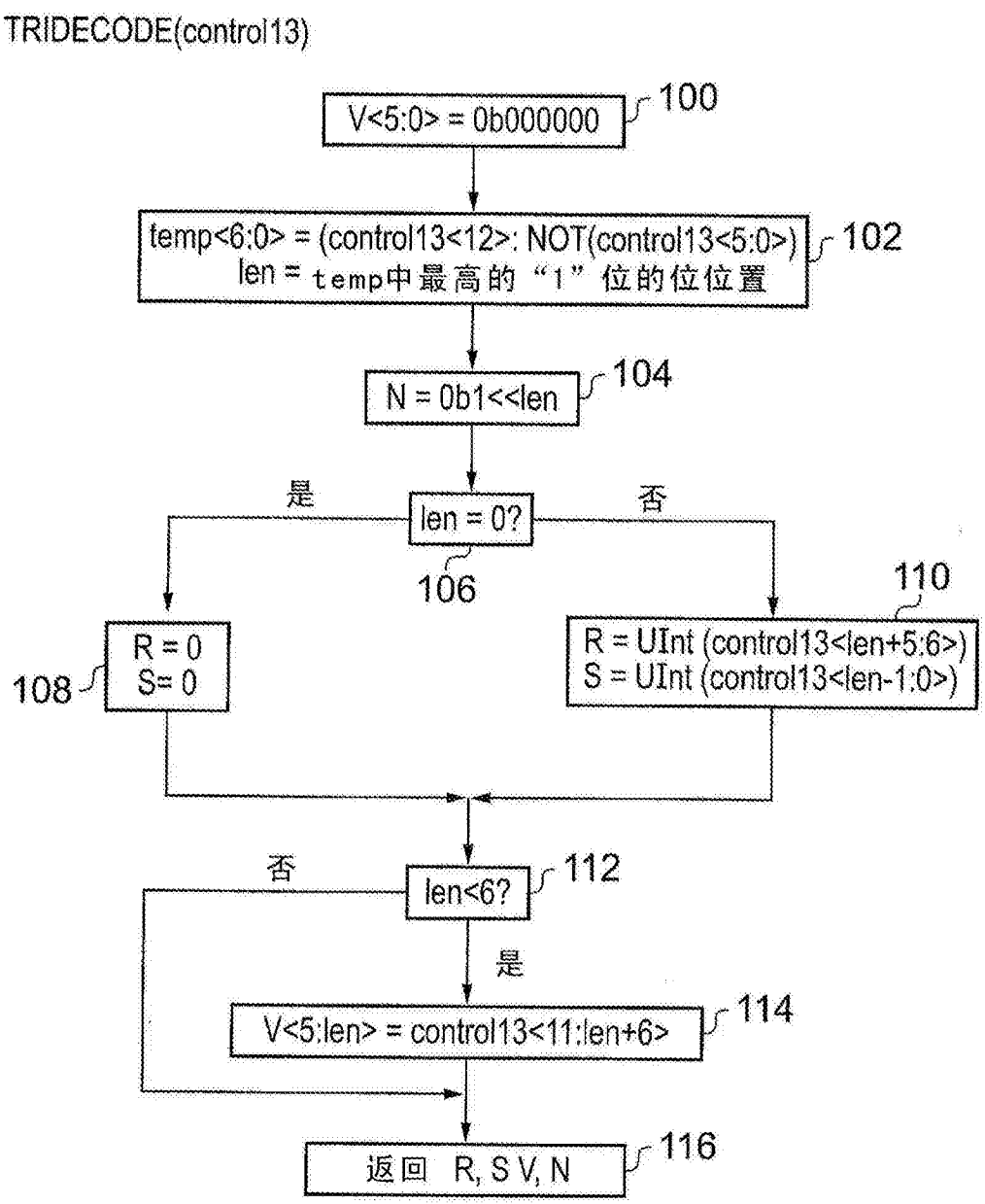

TRIDECODE(control13)
V<5:0> = 0b000000
100
temp<6:0> = (control13<12>: NOT(control13<5:0>)
len = temp中最高的“1”位的位置
102
N = 0b1<<len
104
是
len = 0?
否
106
110
R = 0
S= 0
108
R = UInt (control13<len+5:6>)
S = UInt (control13<len-1:0>)
否
len<6?
112
是
V<5:len> = control13<11:len+6>
114
返回 R, S V, N
116

图10

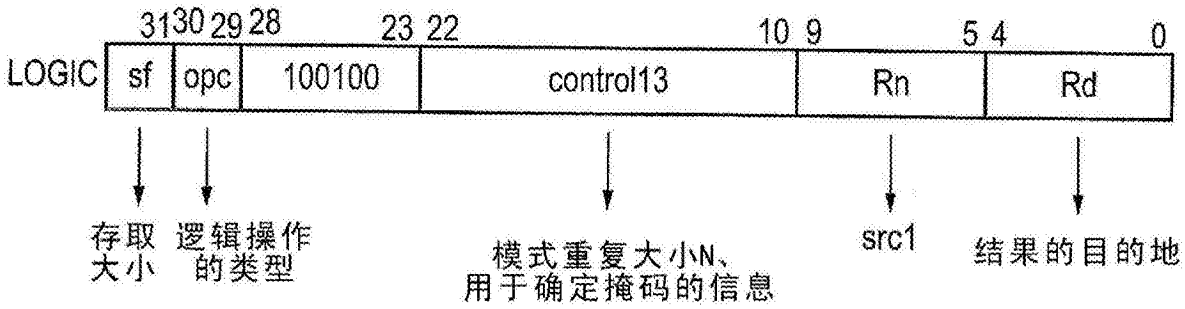

31 30 29 28
23 22
10 9
5 4
0
LOGIC
sf
opc
100100
control13
Rn
Rd
存取大小
逻辑操作的类型
模式重复大小N、用于确定掩码的信息
src1
结果的目的地

图11

Control13 <12:0>:

<12>	<11:6>	<5:0>	模式重复大小N	模式旋转参数R	模式宽度参数S
1	rrrrrr	ssssss	64	UInt(rrrrrr)	UInt(ssssss)
0	xrrrrr	0sssss	32	UInt(rrrrr)	UInt(sssss)
0	xxrrrr	10ssss	16	UInt(rrrr)	UInt(ssss)
0	xxxrrr	110sss	8	UInt(rrr)	UInt(sss)
0	xxxxrr	1110ss	4	UInt(rr)	UInt(ss)
0	xxxxxr	11110s	2	UInt(r)	UInt(s)
0	xxxxxx	111110	1	0	0
0	xxxxxx	111111	未定义		

图12

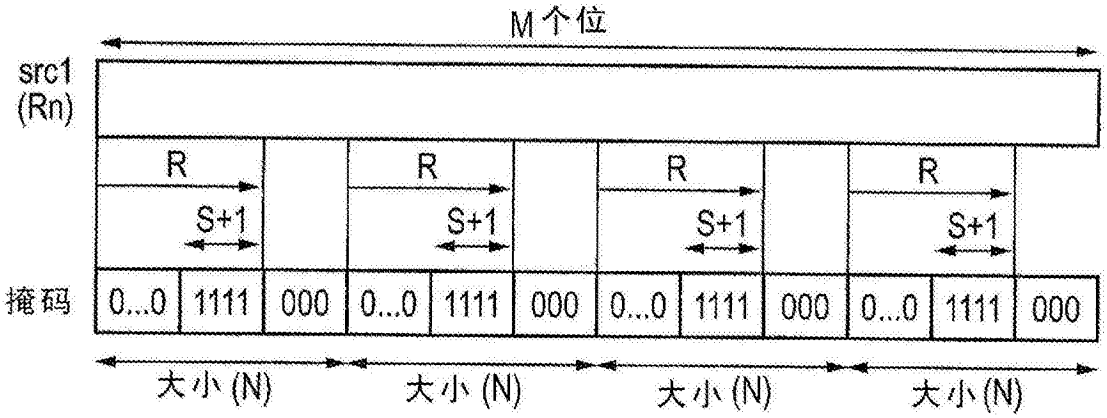


图13

opc = 00 → 结果 = src 与 掩码

opc = 01 → 结果 = src 或 掩码

opc = 10 → 结果 = src异或 掩码

图14

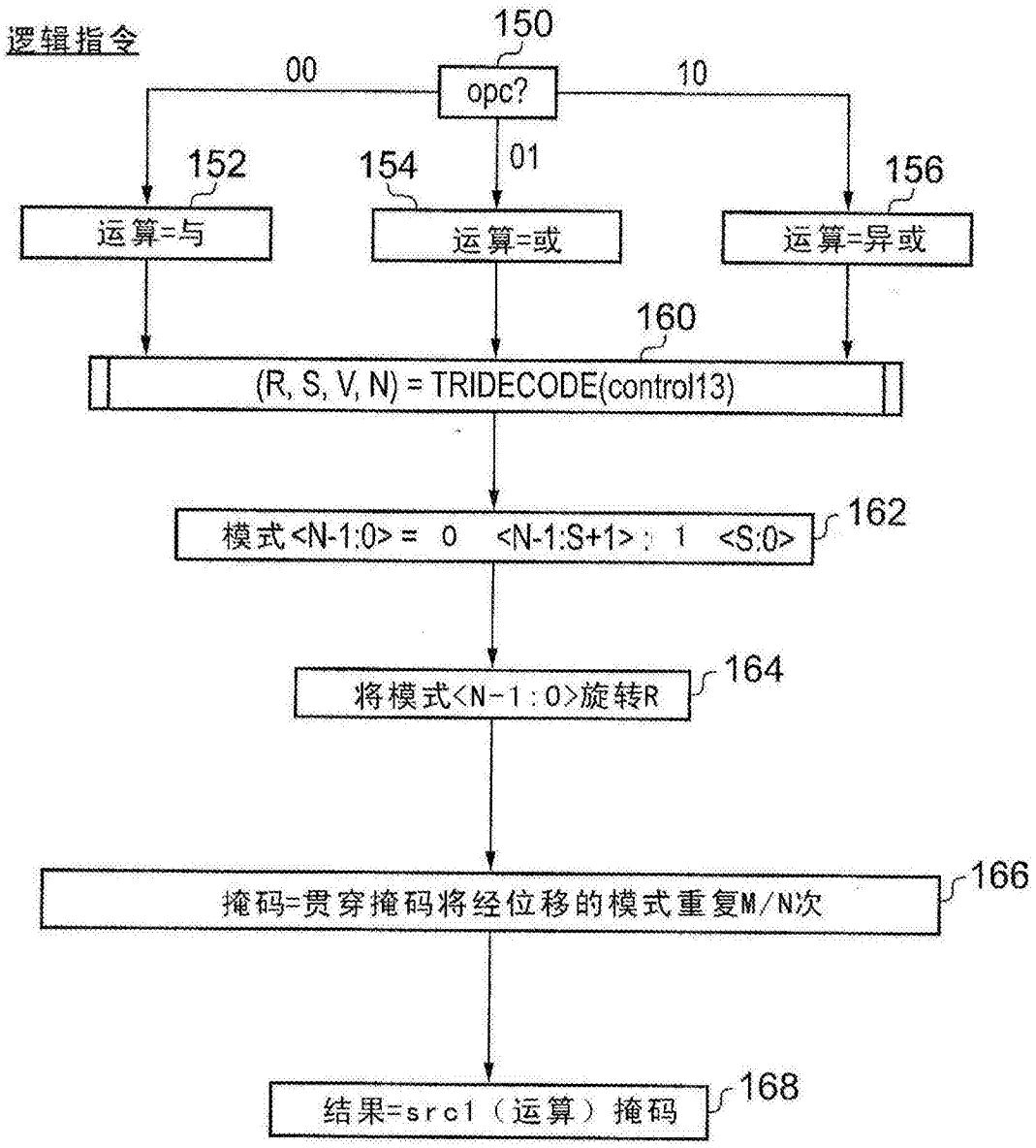

逻辑指令
150
opc?
00
01
10
152
154
156
运算=与
运算=或
运算=异或
160
(R, S, V, N) = TRIDECODE(control13)
模式<N-1:0> = 0 <N-1:S+1> : 1 <S:0>
162
将模式<N-1:0>旋转R
164
掩码=贯穿掩码将经位移的模式重复M/N次
166
结果=src1（运算）掩码
168

图15

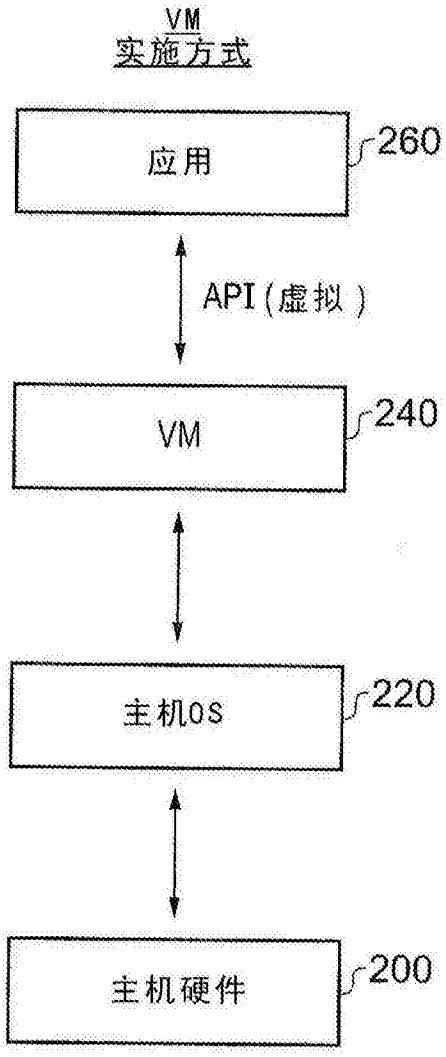


图16