(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 104919432 B
(45)授权公告日 2017.12.22

(21)申请号 201380045387.6
(22)申请日 2013.06.25
(65)同一申请的已公布的文献号
申请公布号 CN 104919432 A
(43)申请公布日 2015.09.16
(30)优先权数据
13/630,131 2012.09.28 US
(85)PCT国际申请进入国家阶段日
2015.02.28
(86)PCT国际申请的申请数据
PCT/US2013/047669 2013.06.25
(87)PCT国际申请的公布数据
WO2014/051782 EN 2014.04.03
(73)专利权人 英特尔公司
地址 美国加利福尼亚州
(72)发明人 M·普罗特尼科夫 I·厄莫拉夫 A·纳赖金 R·凡伦天
(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 何焜
(51)Int.Cl.
G06F 12/02(2006.01)
(56)对比文件
US 2010/0284404 A1,2010.11.11,
US 5832288 A,1998.11.03,
US 2002/0184480 A1,2002.12.05,
US 2012/0078992 A1,2012.03.29,
JP 2010204913 A,2010.09.16,
CN 101154153 A,2008.04.02,
审查员 孟心怡

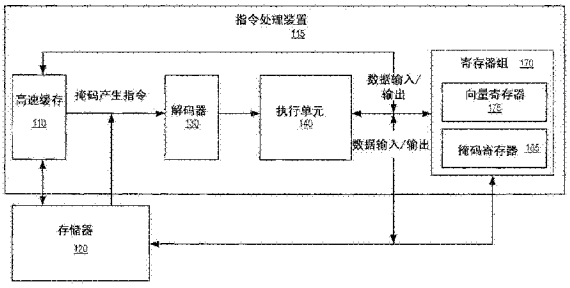
权利要求书2页 说明书13页 附图15页

(54)发明名称

用于将多个位向左移并将多个1拉入较低有效位的指令

(57)摘要

通过处理器执行掩码产生指令，以提高对数据元素数组进行向量操作的效率。处理器包括多个向量寄存器，多个向量寄存器之一存储数组的数据元素。处理器还包括执行电路，该执行电路用于接收掩码产生指令，该掩码产生指令至少指定第一操作数和第二操作数。响应于该掩码产生指令，该执行电路用于将第一操作数的位向左移动在第二操作数中定义的次数，并且每当第一操作数的最高有效位从左边被移出时就从右边拉入位1，以产生结果。该结果中的每个位对应于该数组的多个数据元素中的一个数据元素。



CN 104919432 B

1.一种指令处理装置,包括:

多个向量寄存器,所述多个向量寄存器中的一个向量寄存器用于存储数组的数据元素;以及

解码器,用于接收掩码产生指令,所述掩码产生指令用于至少指定第一操作数和第二操作数,以及

执行单元,耦合至所述多个向量寄存器,所述执行单元用于响应于所述掩码产生指令,将所述第一操作数的位向左移动所述第二操作数中定义的次数,并且用于每当所述第一操作数的最高有效位被移出时拉入最低有效位1,由此产生用于包含多个位的结果,其中所述结果中的每个位用于对应于所述数组中的所述数据元素中的一个数据元素。

2.如权利要求1所述的装置,其特征在于,所述第二操作数用于指定向量操作的剩余循环中的余下迭代的数量。

3.如权利要求2所述的装置,其特征在于,所述第二操作数用于指定所述向量操作的循环限制减去当前迭代计数的减法结果。

4.如权利要求1所述的装置,其特征在于,所述多个向量寄存器包含第一向量寄存器和第二向量寄存器,并且其中所述第二操作数指定用来存储在所述第二向量寄存器中的要合并到所述第一向量寄存器中的现有数据元素中以进行向量计算的数据元素的数量。

5.如权利要求1所述的装置,其特征在于,所述第一操作数和所述第二操作数均是通用寄存器。

6.如权利要求1所述的装置,其特征在于,所述第一操作数是掩码寄存器,且所述第二操作数是通用寄存器。

7.如权利要求1所述的装置,其特征在于,基于所述结果用来设置一个或多个状态寄存器。

8.一种由处理器执行的方法,所述方法包括:

通过处理器来接收掩码产生指令,所述掩码产生指令至少指定第一操作数和第二操作数;以及

响应于所述掩码产生指令,执行以下操作:

将所述第一操作数的位向左移动所述第二操作数中定义的次数,以及

每当所述第一操作数的最高有效位被移出时拉入最低有效位1,由此产生包含多个位的结果,所述结果中的每个位对应于数组的数据元素。

9.如权利要求8所述的方法,其特征在于,所述第二操作数指定向量操作的剩余循环中的余下迭代的数量。

10.如权利要求9所述的方法,其特征在于,所述第二操作数指定所述向量操作的循环限制减去当前迭代计数的减法结果。

11.如权利要求8所述的方法,其特征在于,所述第二操作数指定所述第二向量寄存器中的要合并到所述第一向量寄存器中的现有数据元素中以进行向量计算的数据元素的数量。

12.如权利要求8所述的方法,其特征在于,所述第一操作数和所述第二操作数均是通用寄存器。

13.如权利要求8所述的方法,其特征在于,所述第一操作数是掩码寄存器,且所述第二

操作数是通用寄存器。

14.如权利要求8所述的方法，其特征在于，进一步包括：

基于所述结果修改一个或多个状态寄存器。

15.一种计算机系统，包括：

随机存取存储器；以及

处理器，耦合至所述随机存取存储器，所述处理器包括：

多个向量寄存器，所述多个向量寄存器中的一个向量寄存器用于存储数组的数据元素；以及

解码器，用于接收掩码产生指令，所述掩码产生指令用于至少指定第一操作数和第二操作数，以及

执行单元，耦合至所述多个向量寄存器，所述执行单元用于响应于所述掩码产生指令，将所述第一操作数的位向左移动所述第二操作数中定义的次数，并且用于每当所述第一操作数的最高有效位被移出时拉入最低有效位1，由此产生用于包含多个位的结果，其中所述结果中的每个位用于对应于所述数组中的所述数据元素中的一个数据元素。

16.如权利要求15所述的计算机系统，其特征在于，所述第二操作数用于指定向量操作的剩余循环中的余下迭代的数量。

17.如权利要求15所述的计算机系统，其特征在于，所述多个向量寄存器包含第一向量寄存器和第二向量寄存器，并且其中所述第二操作数用于指定用来存储在所述第二向量寄存器中的要合并到所述第一向量寄存器中的现有数据元素中以进行向量计算的数据元素的数量。

18.如权利要求15所述的计算机系统，其特征在于，所述第一操作数和所述第二操作数均是通用寄存器。

19.如权利要求15所述的计算机系统，其特征在于，所述第一操作数是掩码寄存器，且所述第二操作数是通用寄存器。

20.如权利要求15所述的计算机系统，其特征在于，基于所述结果用来设置一个或多个状态寄存器。

21.一种机器可读存储介质，所述机器可读存储介质包括代码，所述代码在被执行时使机器执行如权利要求8-14中的任一项所述的方法。

22.一种计算机实现的系统，包括用于执行如权利要求8-14中的任一项所述的方法的装置。

用于将多个位向左移并将多个1拉入较低有效位的指令

技术领域

[0001] 本公开涉及处理逻辑、微处理器以及相关联的指令集架构的领域，该指令集架构在被处理器或其它处理逻辑所执行时运行逻辑、数学或其它功能性操作。

背景技术

[0002] 指令集或指令集架构(ISA)是计算机架构中与编程有关的部分，并且可包括原生数据类型、指令、寄存器架构、寻址模式、存储器架构、中断和异常处理、以及外部输入和输出(I/O)。术语指令在本申请中一般表示宏指令，宏指令是被提供给处理器(或指令转换器，该指令转换器(利用静态二进制转换、包括动态编译的动态二进制转换)转换、变形、仿真或以其它方式将指令转换成将由处理器处理的一个或多个其它指令)以供执行的指令——作为对比，微指令或微操作(微操作)是处理器的解码器解码宏指令的结果。

[0003] ISA与微架构不同，微架构是实现该指令集的处理器的内部设计。具有不同微架构的处理器可共享共同的指令集。例如，Intel®酷睿(Core™)处理器、以及来自加利福尼亚州桑尼威尔(Sunnyvale)的超微半导体有限公司(Advanced Micro Devices,Inc.)的诸多处理器实现几乎相同版本的x86指令集(在更新的版本中加入了一些扩展)，但具有不同的内部设计。例如，可利用公知技术(包括专用物理寄存器、利用寄存器重命名机制的一个或多个动态分配的物理寄存器等等)在不同的微架构中以不同的方式实现该ISA的同一寄存器架构。

[0004] 许多现代ISA支持向量操作(也称为紧缩数据操作或单指令多数据(SIMD)操作)。取代仅对一个数据元素或一对数据元素进行操作的标量指令，向量指令(也称为紧缩数据指令或SIMD指令)可同时或并行地对多个数据元素或多对数据元素进行操作。处理器可具有并行的执行硬件，以响应于该向量指令同时或并行地执行多个操作。

[0005] 向量操作在一个操作中对紧缩在一个寄存器或存储器位置之内的多个数据元素进行操作。这些数据元素被称为向量数据元素或紧缩数据元素。每一向量数据元素可表示单独的个体数据片(例如，像素的颜色等)，该数据片可单独地被操作或独立于其他数据被操作。

[0006] 附图简述

[0007] 在附图中的诸个图中通过示例而非限制地示出各个实施例：

[0008] 图1是根据一个实施例的包括向量寄存器和掩码寄存器的指令处理装置的框图。

[0009] 图2A-2C示出根据一个实施例的掩码产生指令的诸个示例。

[0010] 图3A和3B示出根据一个实施例的数组数据对齐的诸个示例。

[0011] 图3C示出根据一个实施例的使用掩码的掩码向量指令的示例。

[0012] 图4示出根据一个实施例的用于给定的向量寄存器宽度和数据元素宽度的掩码位的数量。

[0013] 图5是示出根据一个实施例的用于响应于掩码产生指令而执行的诸个操作的流程图。

[0014] 图6是示出根据一个实施例的使用软件指令转换器将源指令集中的二进制指令转换成目标指令集中的二进制指令的框图。

[0015] 图7A是根据一个实施例的有序和无序流水线的框图。

[0016] 图7B是根据一个实施例的有序和无序核的框图。

[0017] 图8A-B是根据一个实施例的更具体的示例性有序核架构的框图。

[0018] 图9是根据一个实施例的处理器的框图。

[0019] 图10是根据一个实施例的系统的框图。

[0020] 图11是根据一个实施例的第二系统的框图。

[0021] 图12是根据本发明的实施例的第三系统的框图。

[0022] 图13是根据一个实施例的芯片上系统(SoC)的框图。

具体实施方式

[0023] 在以下描述中,陈述了多个具体细节。然而,应当理解的是,可不通过这些具体细节来实施本发明的实施例。在其它实例中,未详细示出公知的电路、结构以及技术,以免模糊对本描述的理解。

[0024] 本申请中描述的诸个实施例提供掩码产生指令,这些掩码产生指令能用于引起或导致处理器产生由经掩码向量指令使用的掩码。经掩码向量指令可应用于如下情形:计算循环的行程计数(即迭代的次数)不能被适配到向量寄存器中的元素的数量所整除。因此,需要单独地处理余下的迭代。为了处理余下的迭代中的元素,掩码产生指令产生适当的断言掩码(predicate mask),该掩码将向量寄存器的部分(例如最高有效位的元素)从计算中忽略或屏蔽,从而将不会产生异常(例如由于访问被分配的在后存储器或/和未定义的结果而引起的异常)。

[0025] 该掩码产生指令也可用于其它情形。例如,该指令可用于更新数据累积中的控制掩码以用于稀疏向量计算。可通过多次迭代执行数据累积。在这些迭代中的一些迭代中,一些数据元素可能退出计算,而一些新数据元素可能加入计算。该控制掩码被更新以跟踪需要进一步计算的元素。可在掩码向量中利用该控制掩码以提高向量计算的效率。

[0026] 与向量指令类似,掩码向量指令能用于引起或导致处理器对一个或多个向量操作数的数据元素执行向量操作。此外,每个掩码向量指令使用掩码来掩码、断言或条件地控制向量操作。掩码能用于按数据元素粒度来掩码或有条件地控制向量处理。例如,掩码能用于对是否将在来自单个源向量操作数的各个数据元素或来自两个源向量操作数的各个相应的数据元素对执行向量操作的结果存储在目的地中进行掩码。掩码向量指令允许对与数据元素单独地且独立地断言或有条件控制的每个数据元素或相应的数据元素对进行向量处理。掩码向量指令、操作和掩码可提供某些优点,诸如增加的代码密度和/或更高的指令吞吐量。

[0027] 图1是指令处理装置115的实施例的框图,该指令处理装置具有执行单元140,该执行单元包括能用于执行指令(包括本申请中描述的掩码产生指令)的电路。在一些实施例中,指令处理装置115可以是处理器、多核处理器的处理器核、或者电子系统中的处理元件。

[0028] 解码器130接收高级机器指令或宏指令形式的传入指令,并且解码所述指令以生成低级微操作、微代码进入点、微指令或其它低级指令或控制信号,它们反映了原始的高级

指令和/或从原始的高级指令导出。低级指令或控制信号可通过低级(例如,电路级或硬件级)操作来实现高级指令的操作。可使用各种不同的机制来实现解码器130。合适机制的示例包括但不限于,微代码、查找表、硬件实现、可编程逻辑阵列(PLA)、用于实现本领域已知的解码器的其它机制等。

[0029] 解码器130可接收针对高速缓存110、存储器120或其它源的传入指令。经解码的指令被发送到执行单元140。执行单元140可从解码器130接收一个或多个微操作、微代码进入点、微指令、其它指令或其它控制信号,它们反映了所接收的指令或者是从所接收的指令导出的。执行单元140从寄存器组170、高速缓存110和/或存储器120接收数据输入并向它们生成数据输出。

[0030] 在一个实施例中,寄存器组170包括架构寄存器,架构寄存器也被称为寄存器。短语"架构寄存器"、"寄存器组"、以及"寄存器"在本文中用于表示对软件和/或编程器可见(例如,软件可见的)和/或由宏指令指定来标识操作数的寄存器,除非另外予以规定或清楚明显可知。这些寄存器不同于给定微架构中的其它非架构式寄存器(例如,临时寄存器、重排序缓冲器、引退寄存器等)。

[0031] 或者,在一个或多个其它实施例中,并非具有解码器130,指令处理装置115可替代地具有指令仿真器、转换器、变形器(morpher)、解释器或者其他指令变换逻辑。各种不同类型的指令变换逻辑在本领域中是已知的,并且可在软件、硬件、固件、或者其组合中实现。指令变换逻辑可接收多个掩码产生指令中的一个或多个,并且仿真、转换、变形、解释、或者以其他方式将指令变换成一个或多个对应的导出指令或控制信号。在又一个其它实施例中,指令处理装置115可既具有解码器又具有附加的指令变换逻辑。例如,指令处理装置115可具有指令变换逻辑和解码器,该指令变换逻辑用于将多个掩码产生指令中的一个或多个变换成一个或多个中间指令,该解码器用于将一个或多个中间指令解码成能由指令处理装置的原生硬件执行的一个或多个较低级的指令或控制信号。指令变换逻辑中的一些或全部可位于指令处理装置的余下部分的管芯外,诸如在单独的管芯上或在管芯外的存储器中。

[0032] 根据一个实施例,寄存器组170包括一组向量寄存器175和一组掩码寄存器185,向量寄存器和掩码寄存器都可用于存储掩码产生指令的操作数。每个向量寄存器175可以是512位、256位或128位宽,或者可以使用不同的向量宽度。每个掩码寄存器185包含多个掩码位,其中每个掩码位对应于多个向量寄存器175中的一个向量寄存器的一个数据元素。由于每个掩码位用于对向量寄存器的数据元素进行掩码,所以64位的掩码寄存器可用于对512位寄存器的六十四个8位数据元素进行掩码。对于具有不同宽度(例如256位或128位)的向量寄存器和不同尺寸(例如16位、32位或64位)的数据元素,可结合向量操作来使用不同数量的掩码位。

[0033] 为了避免混淆描述,已示出和描述了相对简单的指令处理装置115。应当理解,其它实施例可具有超过一个执行单元。例如,装置115可包括多个不同类型的执行单元,诸如例如算术单元、算术逻辑单元(ALU)、整数单元、浮点单元等。指令处理装置或处理器的再其它实施例可具有多个核、逻辑处理器或执行引擎。稍后将参考图7-13提供指令处理装置115的多个实施例。

[0034] 根据本发明的实施例,本申请中描述的掩码产生指令通过移动该指令的寄存器操作数中的多个位来产生掩码。该寄存器操作数可以是掩码寄存器或通用寄存器。图2A-2C示

出用于多个掩码产生指令的伪代码的示例。在这些附图中，r1、r2表示独立尺寸的通用寄存器(例如r1可以是32位，而r2是64位)，并且k1表示掩码寄存器。值KL表示多个掩码位的数量，该值KL可根据附加至该指令末尾的助记符B/W/D/Q而确定。

[0035] 图2A示出掩码产生指令KSHLONES[B/W/D/Q]k1,r2的示例。助记符B/W/D/Q表示指令KSHLONES具有四种形式：KSHLONESB、KSHLONESW、KSHLONESD以及KSHLONESQ，分别对应于8位、16位、32位、64位的掩码。在本示例中，k1掩码既担当源操作数又担当目的地。另一源操作数是通用寄存器或来自存储器的值。

[0036] KSHLONES指令将k1掩码的多个位向左移动在源操作数(r2或存储器)中定义的次数，并拉入多个1以填充较低位的位置。本申请中的术语“向左移”或“左移”表示按照从最低有效位(LSB)到最高有效位(MSB)的方向移动位。即，每当k1掩码被向左移动一个位位置时，将位值1拉入以填充最低有效位位置。例如，如果k1＝1:0:0:0:1:1:0:0且r2＝4，则″KSHLONESBk1,r2″将产生结果k1＝1:1:0:0:1:1:1:1，其中每个“0”和“1”表示位值。应注意到所得的(目的地)k1中保留的那些k1位仅仅在位置上被移动，并且它们的值未被该移动改变。添加至LSB位置的新的位都是1。

[0037] 图2B示出掩码产生指令SHLONES[B/W/D/Q]r1,r2的替代实施例，该指令使用通用寄存器r1既作为源操作数又作为目的地。该形式的指令实现其作为互补位操纵指令的用途。图2C示出掩码产生指令的另一替代实施例，该掩码产生指令修改状态标志(ZF,CF)，使得该指令能被直接用于控制流。掩码产生指令的另一实施例将经移动的结果(即所得的掩码)存储至与源操作数不同的目的地寄存器；例如KSHLONES k1,k2,r2和SHLONES r1,r2,r3。可能存在这些指令的附加替代实施例，这些附加替代实施例不一定具有与上述掩码产生指令相同的指令格式。在以下描述中，掩码产生指令的各种形式被称为KSHLONES及其变型。

[0038] 图3A和3B示出多个示例情形，其中可使用KSHLONES及其变型来提高向量计算的效率。在这些示例中，向量操作的剩余循环中的剩余数组元素未填满整个向量寄存器。在这些元素中，假定向量寄存器可存储多达16个数组元素：例如，该向量寄存器具有512位，而每个数组元素是32位双字。如果数组元素的总数是35且循环的开始与向量寄存器对齐(如图3A中所示)，则在结束时将有三个剩余数据元素未在向量化循环中被处理且需要单独地被处理。如果数组元素的总数是35并且该循环的开始未与向量寄存器对齐(如图3B中所示的第一向量化循环中的两个数组元素)，在结束时将有一个剩余数组元素未在该向量化循环中被处理并且需要单独地被处理。本申请中描述的掩码产生指令产生掩码，该掩码能与剩余数组元素一起在掩码向量操作中使用以改进循环向量化。

[0039] 为了提高数据访问的效率，编译器可产生代码以用于单独地处理最后向量化循环中的剩余数组元素。然而，由于数组元素的地址和/或循环行程计数在编译时未知，所以一般不能在编译时求解最后向量化循环中的数组元素的数量。利用本申请中描述的实施例，编译器可在编译时产生一个或多个掩码产生指令来代替执行相同任务的其它代码序列。因此，编译器可利用这些掩码产生指令来简化其循环优化任务。在替代实施例中，这些掩码产生指令可由编程器或其它代码产生实体使用。

[0040] KSHLONES指令及其变型可用于处理如下情形：在循环最后的剩余数据元素的总尺寸小于向量寄存器的宽度。这意味着，当循环中没有足够多的迭代(且数组中没有足够多的

数据元素)来构成全宽度向量操作时,可使用KSHLONES指令及其变型。

[0041] 在图3C的示例中,该数组的最后三个数据元素(即,A(32)、A(33)、A(34))未占据源向量307的全部宽度。即,在A中余下的元素不足以填满整个向量寄存器。因为源向量307包含A(32)、A(33)、A(34)作为其最低阶数据元素,因此掩码308的仅最低阶三个位被置位(例如置为1),以指示对于A(32)、A(33)、A(34)应执行加法,且应当存储该加法的结果。该掩码308的较高阶13个位被清零(例如0)。掩码308可以是由处理器执行指令KSHLONES或其变型之一而生成的结果。

[0042] 在一个实施例中,在数组的末尾缺少数据元素(用于填满整个向量寄存器)可以是在数组的基址处最初未对齐的结果。例如,在图像处理应用中,通常图像数组的尺寸是向量寄存器宽度的整数倍。然而,如果图像数组的开始未被对齐,则在该循环结束时可能余下不能填满整个向量寄存器的多个数据元素。

[0043] 使用掩码308有助于对其中数组的数据元素是操作数的循环的执行进行向量化。在图3C的示例中,可利用其中源向量307与掩码308一起使用的掩码向量操作来对迭代索引i=32、33和34进行向量化。在一个实施例中,在检测到循环之后,编译器可产生循环优化代码,该循环优化代码包含本申请中描述的一个或多个掩码产生指令。

[0044] 用于所说明的掩码向量操作303的指令指示要被加到标量值的源向量。其它掩码向量指令可指示两个或超过两个源向量。掩码向量操作303的指令还指定掩码308。多个掩码中的每一个掩码可各自包括多个掩码元素、断言元素、条件控制元素或标志。如图所示,在涉及一个源向量操作数的操作的情况下,对于每个对应的源数据元素可以有一个这样的掩码元素或标志。通常,每个元素或标志可以是单个位。单个位可允许指定两个不同的可能性(例如,执行操作或不执行操作,存储操作的结果或不存储操作的结果,等等)中的任一个。替代地,如果需要在超过两个不同选项之间进行选择,则可对每个标志或元素使用两个或超过两个位。

[0045] 根据所示的协定,当给定的掩码位被置位为一时,对源向量的相应数据元素执行向量操作的结果,并且将该结果存储在结果的相应数据元素中。相反,当给定的掩码位被清零时,则对于源向量的相应数据元素忽略(即不执行)该向量操作,或者不允许将结果存储在结果的相应数据元素中。相反,可将另一个值存储在结果数据元素中。例如,存储来自源向量的相应数据元素的数值。在替代实施例中,可将零或另一个预定值存储在结果的相应数据元素中。与所示相反的协定也是可能的,其中多个位被清零(即,0)以允许存储结果,或被置位(即,1)以不允许存储结果。

[0046] 以下示例代码序列产生用于剩余循环的掩码,其中当前迭代计数被存储在rbx中,且循环限制被存储在rex中。利用图3C的所示实施例,当前迭代计数是31,且循环限制是34。

[0047] SUB rbx,rcx//计算余下迭代的数量

[0048] KXOR k1,k1,k1//清零掩码

[0049] KSHLONES k1,rbx//产生用于剩余循环的掩码

[0050] 使用KSHLONES指令(包括其变型)来产生用于剩余循环的掩码存在许多优势。KSHLONES指令可利用减法结果进行操作。对于包含减法作为其操作的部分的另一指令,将会引发附加的预计算开销,以用于在减法之前执行操作数类型比较。此外,KSHLONES指令覆盖了迭代计数器和/或循环限制会是负值的情形,这允许编译器优化代码的更多的可变性。

附加地，用于产生用于剩余循环的掩码的代码被划分成三个阶段(即上述代码序列中的三个指令)，这改进了执行调度并提供使用KSHLONES指令时的更多可变性和灵活性。KSHLONES指令可自身被使用，或在不需要操作数相减的场合与其它指令组合地使用。例如，当1的数量(N)已知时，KSHLONES可用于如下地产生具有最低有效N位中的多个1的掩码：N＝5；k1＝0:0:0:0:0:0:0:0;KSHLONES k1,N导致k1＝0:0:0:1:1:1:1:1。

[0051] KSHLONES指令还可在数据累积中使用以用于稀疏向量计算，如图4的示例中所示。在该示例中，利用一对向量寄存器(V1和V2)和一对掩码寄存器(K1和K2)来执行数据累积。V1和V2均是稀疏向量，其中并非所有的数据元素位置都被填充。V1担当累积器以累积用于计算的向量元素，而V2提供新数据元素以填充在V1的未利用的槽中。掩码寄存器K1和K2用于指示相应向量寄存器中的哪些位置包含用于计算的有效数据元素。在本示例中，对于K1和K2，与有效数据元素相对应的掩码位被设置为1。应理解，可将K2的位值预留给V2的相同数据元素。

[0052] 在图4的示例中，V2最初包含被指示为B0的四个元素。K2中的相应掩码位指示这四个元素的位置。通过使用N＝POPCNT(K2)，N的值被设置为具有值1的K2位的数量。因此，在本示例中，N＝4。K1中的掩码位包含与最初V1的元素位置0-2相对应的三个1。K1中包含的信息不仅指示累积的元素A0的数量，而且指示V1内的空槽的右边界(在本示例中，该右边界在第三元素位置处)。K1可被原样地或取反地使用以进行进一步的数据累积，进一步的数据累积包括COMPRESS和/或EXPAND指令。

[0053] 利用现有的向量指令，可将四个B0压缩和合并至V1的元素位置3-6中。经更新的V1变得比初始V1更紧密，因此更易于实现高效的向量计算。可通过K1＝KSHLONES(K1,N)来计算合并之后的相应K1，其保留K1的源值中的最初三个位1并加上四个附加的位1。保留K1的源值消除了保持单独的计数器以在合并之前和之后跟踪累积器内的元素的数量的需要。在经更新的V1用于向量计算之后，可重复图4的操作，以使累积器可继续累积用于向量计算的数据元素。

[0054] 本申请公开的掩码产生指令是具有一般用途的通用指令。例如，可单独地或与其它指令组合地使用这些指令，以计算用于向量操作的剩余循环或用于稀疏向量计算中的数据累积的掩码。基于本公开还可构想其它使用。

[0055] 图5是根据一个实施例的用于执行掩码产生指令的方法500的流程框图。方法500开始于处理器(更具体地，例如图1的执行单元140)接收掩码产生指令，该掩码产生指令指定至少第一操作数和第二操作数(框510)。掩码产生指令的示例包括以上描述的KSHLONES指令及其变型。在一个实施例中，第一操作数是掩码寄存器，而第二操作数是通用寄存器。在替代实施例中，第一操作数和第二操作数均是通用寄存器。响应于该掩码产生指令，处理器执行以下操作(框520)：将第一操作数的位向左移动第二操作数中定义的次数(框530)，并且每当第一操作数的最高有效位被移出(向左)时拉入最低有效位1，由此产生结果(框540)。该结果中的每个位对应于数据元素。该结果是用于掩码向量操作的掩码。

[0056] 在各实施例中，方法500的方法可由通用处理器、专用处理器(例如，图形处理器或数字信号处理器)、或另一种类型的数字逻辑设备或指令处理装置执行。在一些实施例中，方法500可由图1的指令处理装置115、或类似的处理器、装置或系统(诸如图7-13中所示的实施例)执行。而且，图1的指令处理装置115以及图7-13中所示的处理器、装置或系统可执

行与方法500的实施例相同、类似或不同的操作和方法的实施例。

[0057] 在一些实施例中，图1的指令处理装置115可结合指令转换器操作，该指令转换器将来自源指令集的指令转换到目标指令集。例如，指令转换器可以变换(例如使用静态二进制变换、包括动态编译的动态二进制变换)、变形、仿真或以其它方式将指令转换成将由核来处理的一个或多个其它指令。指令转换器可以用软件、硬件、固件、或其组合实现。指令转换器可以在处理器上、在处理器外、或者部分在处理器上且部分在处理器外。

[0058] 图6是对比根据本发明的实施例的软件指令转换器的使用的框图。在所示的实施例中，指令转换器是软件指令转换器，但作为替代，该指令转换器可以用软件、固件、硬件或其各种组合来实现。图6示出可以使用x86编译器604来编译利用高级语言602的程序，以生成可以由具有至少一个x86指令集核的处理器616原生执行的x86二进制代码606。具有至少一个x86指令集核的处理器616表示任何处理器，这些处理器能通过兼容地执行或以其它方式处理以下内容来执行与具有至少一个x86指令集核的英特尔处理器基本相同的功能：1)英特尔x86指令集核的指令集的本质部分，或2)目标为在具有至少一个x86指令集核的英特尔处理器上运行的应用或其它程序的目标代码版本，以便取得与具有至少一个x86指令集核的英特尔处理器基本相同的结果。x86编译器604表示用于生成x86二进制代码606(例如，目标代码)的编译器，该二进制代码可通过或不通过附加的链接处理在具有至少一个x86指令集核的处理器616上执行。类似地，图6示出可以使用替代的指令集编译器608来编译利用高级语言602的程序，以生成可以由不具有至少一个x86指令集核的处理器614(例如具有执行加利福尼亚州桑尼维尔市的MIPS技术公司的MIPS指令集、和/或执行加利福尼亚州桑尼维尔市的ARM控股公司的ARM指令集的核的处理器)原生执行的替代指令集二进制代码610。指令转换器612被用来将x86二进制代码606转换成可以由不具有x86指令集核的处理器614原生执行的代码。该转换后的代码不大可能与替代性指令集二进制代码610相同，因为能够这样做的指令转换器难以制造；然而，转换后的代码将完成一般操作并由来自替代指令集的指令构成。因此，指令转换器612通过仿真、模拟或任何其它过程来表示允许不具有x86指令集处理器或核的处理器或其它电子设备执行x86二进制代码606的软件、固件、硬件或其组合。

[0059] 示例性核架构

[0060] 有序和无序核框图

[0061] 图7A是示出根据本发明的各实施例的示例性有序流水线和示例性的寄存器重命名的无序发布/执行流水线的框图。图7B是示出根据本发明的各实施例的要包括在处理器中的有序架构核的示例性实施例和示例性的寄存器重命名的无序发布/执行架构核的框图。图7A和7B中的实线框示出了有序流水线和有序核，而可选增加的虚线框示出了寄存器重命名的、无序发布/执行流水线和核。给定有序方面是无序方面的子集的情况下，将描述无序方面。

[0062] 在图7A中，处理器流水线700包括取出级702、长度解码级704、解码级706、分配级708、重命名级710、调度(也称为分派或发布)级712、寄存器读取/存储器读取级714、执行级716、写回/存储器写入级718、异常处理级722和提交级724。

[0063] 图7B示出了包括耦合到执行引擎单元750的前端单元730的处理器核790，且执行引擎单元和前端单元两者都耦合到存储器单元770。核790可以是精简指令集计算(RISC)

核、复杂指令集计算(CISC)核、超长指令字(VLIW)核或混合或替代核类型。作为又一选项，核790可以是专用核，诸如例如网络或通信核、压缩引擎、协处理器核、通用计算图形处理器单元(GPGPU)核、或图形核等等。

[0064] 前端单元730包括耦合到指令高速缓存单元734的分支预测单元732，该指令高速缓存单元耦合到指令转换后备缓冲器(TLB)736，该指令转换后备缓冲器耦合到指令取出单元738，指令取出单元耦合到解码单元740。解码单元740(或解码器)可解码指令，并生成从原始指令解码出的、或以其它方式反映原始指令的、或从原始指令导出的一个或多个微操作、微代码进入点、微指令、其它指令、或其它控制信号作为输出。解码单元740可使用各种不同的机制来实现。合适的机制的示例包括但不限于查找表、硬件实现、可编程逻辑阵列(PLA)、微代码只读存储器(ROM)等。在一个实施例中，核790包括(例如，在解码单元740中或否则在前端单元730内的)用于存储某些宏指令的微代码的微代码ROM或其它介质。解码单元740耦合至执行引擎单元750中的重命名/分配器单元752。

[0065] 执行引擎单元750包括重命名/分配器单元752，该重命名/分配器单元耦合至引退单元754和一个或多个调度器单元756的集合。调度器单元756表示任何数目的不同调度器，包括预留站、中央指令窗等。调度器单元756耦合到物理寄存器组单元758。每个物理寄存器组单元758表示一个或多个物理寄存器组，其中不同的物理寄存器组存储一种或多种不同的数据类型，诸如标量整数、标量浮点、紧缩整数、紧缩浮点、向量整数、向量浮点、状态(例如，作为要执行的下一指令的地址的指令指针)等。在一个实施例中，物理寄存器组单元758包括向量寄存器单元、写掩码寄存器单元和标量寄存器单元。这些寄存器单元可以提供架构向量寄存器、向量掩码寄存器、和通用寄存器。物理寄存器组单元758与引退单元754重叠以示出可以用来实现寄存器重命名和无序执行的各种方式(例如，使用重新排序缓冲器和引退寄存器组；使用将来的文件、历史缓冲器和引退寄存器组；使用寄存器映射和寄存器池等等)。引退单元754和物理寄存器组单元758耦合到执行群集760。执行群集760包括一个或多个执行单元762的集合和一个或多个存储器访问单元764的集合。执行单元762可以对各种类型的数据(例如，标量浮点、紧缩整数、紧缩浮点、向量整型、向量浮点)执行各种操作(例如，移位、加法、减法、乘法)。尽管一些实施例可以包括专用于特定功能或功能集合的多个执行单元，但其它实施例可包括全部执行所有功能的仅一个执行单元或多个执行单元。调度器单元756、物理寄存器组单元758、执行群集760被示出为可能是复数个，因为某些实施例为某些数据/操作类型创建了诸个单独流水线(例如，均具有各自调度器单元、物理寄存器组单元和/或执行群集的标量整数流水线、标量浮点/紧缩整数/紧缩浮点/向量整数/向量浮点流水线、和/或存储器访问流水线，以及在单独的存储器访问流水线的情况下特定实施例被实现为仅仅该流水线的执行群集具有存储器访问单元764)。还应当理解，在使用分开的流水线的情况下，这些流水线中的一个或多个可以为无序发布/执行，并且其余流水线可以为有序发布/执行。

[0066] 存储器访问单元764的集合耦合到存储器单元770，该存储器单元包括耦合到数据高速缓存单元774的数据TLB单元772，其中数据高速缓存单元耦合到二级(L2)高速缓存单元776。在一个示例性实施例中，存储器访问单元764可包括加载单元、存储地址单元和存储数据单元，其中的每一个均耦合至存储器单元770中的数据TLB单元772。指令高速缓存单元734还耦合到存储器单元770中的第二级(L2)高速缓存单元776。L2高速缓存单元776耦合到

一个或多个其它级的高速缓存，并最终耦合到主存储器。

[0067] 作为示例，示例性寄存器重命名的、无序发布/执行核架构可以如下实现流水线700：1)指令取出738执行取出和长度解码级702和704；2)解码单元740执行解码级706；3)重命名/分配器单元752执行分配级708和重命名级710；4)调度器单元756执行调度级712；5)物理寄存器组单元758和存储器单元770执行寄存器读取/存储器读取级714；执行群集760执行执行级716；6)存储器单元770和物理寄存器组单元758执行写回/存储器写入级718；7)各单元可牵涉到异常处理级722；以及8)引退单元754和物理寄存器组单元758执行提交级724。

[0068] 核790可支持一个或多个指令集(例如，x86指令集(具有与较新版本一起添加的一些扩展)；加利福尼亚州桑尼维尔市的MIPS技术公司的MIPS指令集；加利福尼州桑尼维尔市的ARM控股的ARM指令集(具有诸如NEON等可选附加扩展))，其中包括本文中描述的各指令。在一个实施例中，核790包括用于支持紧缩数据指令集扩展(例如SSE、AVX1、AVX2等等)的逻辑，由此允许许多多媒体应用所使用的操作利用紧缩数据来执行。

[0069] 应当理解，核可支持多线程化(执行两个或更多个并行的操作或线程的集合)，并且可以按各种方式来完成该多线程化，此各种方式包括时分多线程化、同步多线程化(其中单个物理核为物理核正在同步多线程化的各线程中的每一个线程提供逻辑核)、或其组合(例如，时分取出和解码以及此后诸如用Intel®超线程化技术来同步多线程化)。

[0070] 尽管在无序执行的上下文中描述了寄存器重命名，但应当理解，可以在有序架构中使用寄存器重命名。尽管所示出的处理器的实施例还包括分开的指令和数据高速缓存单元734/774以及共享L2高速缓存单元776，但替代实施例可以具有用于指令和数据两者的单个内部高速缓存，诸如例如一级(L1)内部高速缓存或多个级别的内部高速缓存。在一些实施例中，该系统可包括内部高速缓存和在核和/或处理器外部的外部高速缓存的组合。或者，所有高速缓存都可以在核和/或处理器的外部。

[0071] 具体的示例性有序核架构

[0072] 图8A-B示出了更具体的示例性有序核架构的框图，该核将是芯片中的若干逻辑块之一(包括相同类型和/或不同类型的其它核)。根据应用，这些逻辑块通过高带宽的互连网络(例如，环形网络)与一些固定的功能逻辑、存储器I/O接口和其它必要的I/O逻辑通信。

[0073] 图8A是根据本发明的各实施例的单个处理器核以及它与管芯上互连网络802的连接及其二级(L2)高速缓存的本地子集804的框图。在一个实施例中，指令解码器800支持具有紧缩数据指令集扩展的x86指令集。L1高速缓存806允许对进入标量和向量单元中的高速缓存存储器的低等待时间访问。尽管在一个实施例中(为了简化设计)，标量单元808和向量单元810使用分开的寄存器集合(分别为标量寄存器812和向量寄存器814)，并且在这些寄存器之间转移的数据被写入到存储器并随后从一级(L1)高速缓存806读回，但是本发明的替代实施例可以使用不同的方法(例如使用单个寄存器集合或包括允许数据在这两个寄存器组之间传输而无需被写入和读回的通信路径)。

[0074] L2高速缓存的本地子集804是全局L2高速缓存的一部分，该全局L2高速缓存被划分成多个分开的本地子集，即每个处理器核一个本地子集。每个处理器核具有到其自己的L2高速缓存804的本地子集的直接访问路径。被处理器核读出的数据被存储在其L2高速缓存子集804中，并且可以与其它处理器核访问其自己的本地L2高速缓存子集并行地被快速

访问。被处理器核写入的数据被存储在其自己的L2高速缓存子集804中，并在必要的情况下从其它子集清除。环形网络确保共享数据的一致性。环形网络是双向的，以允许诸如处理器核、L2高速缓存和其它逻辑块之类的代理在芯片内彼此通信。每个环形数据路径为每个方向1012位宽。

[0075] 图8B是根据本发明的各实施例的图8A中的处理器核的一部分的展开图。图8B包括L1高速缓存804的L1数据高速缓存806A部分，以及关于向量单元810和向量寄存器814的更多细节。具体地说，向量单元810是16宽向量处理单元(VPU)(见16宽ALU 828)，该单元执行整型、单精度浮点以及双精度浮点指令中的一个或多个。该VPU通过混合单元820支持对寄存器输入的混合、通过数值转换单元822A-B支持数值转换、并通过复制单元824支持对存储器输入的复制。写掩码寄存器826允许断言所得的向量写入。

[0076] 具有集成存储器控制器和图形器件的处理器

[0077] 图9是根据本发明的各实施例可能具有多于一个核、可能具有集成存储器控制器、以及可能具有集成图形器件的处理器900的框图。图9中的实线框示出具有单个核902A、系统代理910、一个或多个总线控制器单元916的集合的处理器900，而虚线框的可选附加示出具有多个核902A-N、系统代理单元910中的一个或多个集成存储器控制器单元914的集合以及专用逻辑908的替代处理器900。

[0078] 因此，处理器900的不同实现可包括：1)CPU，其中专用逻辑908是集成图形和/或科学(吞吐量)逻辑(其可包括一个或多个核)，并且核902A-N是一个或多个通用核(例如，通用的有序核、通用的无序核、这两者的组合)；2)协处理器，其中核902A-N是旨在主要用于图形和/或科学(吞吐量)的多个专用核；以及3)协处理器，其中核902A-N是多个通用有序核。因此，处理器900可以是通用处理器、协处理器或专用处理器，诸如例如网络或通信处理器、压缩引擎、图形处理器、GPGPU(通用图形处理单元)、高吞吐量的集成众核(MIC)协处理器(包括30个或更多核)、或嵌入式处理器等。该处理器可以被实现在一个或多个芯片上。处理器900可以是一个或多个衬底的一部分，和/或可以使用诸如例如BiCMOS、CMOS或NMOS等的多个加工技术中的任何一个技术将该处理器实现在一个或多个衬底上。

[0079] 存储器层次结构包括在各核内的一个或多个级别的高速缓存、一个或多个共享高速缓存单元906的集合、以及耦合至集成存储器控制器单元914的集合的外部存储器(未示出)。该共享高速缓存单元906的集合可以包括一个或多个中间级高速缓存，诸如二级(L2)、三级(L3)、四级(L4)或其它级别的高速缓存、末级高速缓存(LLC)、和/或其组合。尽管在一个实施例中，基于环的互连单元912将集成图形逻辑908、共享高速缓存单元906的集合以及系统代理单元910/集成存储器控制器单元914互连，但替代实施例可使用任何数量的公知技术来将这些单元互连。在一个实施例中，可以维护一个或多个高速缓存单元906和核902-A-N之间的一致性(coherency)。

[0080] 在一些实施例中，核902A-N中的一个或多个核能够多线程化。系统代理910包括协调和操作核902A-N的那些组件。系统代理单元910可包括例如功率控制单元(PCU)和显示单元。PCU可以是或包括用于调整核902A-N和集成图形逻辑908的功率状态所需的逻辑和组件。显示单元用于驱动一个或多个外部连接的显示器。

[0081] 核902A-N在架构指令集方面可以是同构的或异构的；即，这些核902A-N中的两个或更多个核可能能够执行相同的指令集，而其它核可能能够执行该指令集的仅仅子集或不

同的指令集。

[0082] 示例性计算机架构

[0083] 图10-13是示例性计算机架构的框图。本领域已知的对膝上型设备、台式机、手持PC、个人数字助理、工程工作站、服务器、网络设备、网络集线器、交换机、嵌入式处理器、数字信号处理器(DSP)、图形设备、视频游戏设备、机顶盒、微控制器、蜂窝电话、便携式媒体播放器、手持设备以及各种其它电子设备的其它系统设计和配置也是合适的。一般地,能够包含本文中所公开的处理器和/或其它执行逻辑的多个系统和电子设备一般都是合适的。

[0084] 现在参考图10,所示出的是根据本发明一个实施例的系统1000的框图。系统1000可以包括一个或多个处理器1010、1015,这些处理器耦合到控制器中枢1020。在一个实施例中,控制器中枢1020包括图形存储器控制器中枢(GMCH)1090和输入/输出中枢(IOH)1050(其可以在分开的芯片上);GMCH 1090包括存储器和图形控制器,存储器1040和协处理器1045耦合到该存储器和图形控制器;IOH 1050将输入/输出(I/O)设备1060耦合到GMCH 1090。或者,存储器和图形控制器中的一个或两者被集成在处理器内(如本文中所描述的),存储器1040和协处理器1045直接耦合到处理器1010以及控制器中枢1020,该控制器中枢与IOH 1050处于单个芯片中。

[0085] 附加处理器1015的任选性质用虚线表示在图10中。每一处理器1010、1015可包括本文中描述的处理核中的一个或多个,并且可以是处理器900的某一版本。

[0086] 存储器1040可以是例如动态随机存取存储器(DRAM)、相变存储器(PCM)或这两者的组合。对于至少一个实施例,控制器中枢1020经由诸如前端总线(FSB)之类的多分支总线、诸如快速通道互连(QPI)之类的点对点接口、或者类似的连接1095与处理器1010、1015进行通信。

[0087] 在一个实施例中,协处理器1045是专用处理器,诸如例如高吞吐量MIC处理器、网络或通信处理器、压缩引擎、图形处理器、GPGPU、或嵌入式处理器等等。在一个实施例中,控制器中枢1020可以包括集成图形加速器。

[0088] 在物理资源1010、1015之间会存在包括架构、微架构、热、和功耗特征等的一系列品质度量方面的各种差异。

[0089] 在一个实施例中,处理器1010执行控制一般类型的数据处理操作的指令。协处理器指令可嵌入在这些指令中。处理器1010将这些协处理器指令识别为应当由附连的协处理器1045执行的类型。因此,处理器1010在协处理器总线或者其它互连上将这些协处理器指令(或者表示协处理器指令的控制信号)发布到协处理器1045。协处理器1045接受并执行所接收的协处理器指令。

[0090] 现在参考图11,所示为根据本发明的一实施例的更具体的第一示例性系统1100的框图。如图11所示,多处理器系统1100是点对点互连系统,并包括经由点对点互连1150耦合的第一处理器1170和第二处理器1180。处理器1170和1180中的每一个都可以是处理器900的某一版本。在本发明的一个实施例中,处理器1170和1180分别是处理器1010和1015,而协处理器1138是协处理器1045。在另一实施例中,处理器1170和1180分别是处理器1010和协处理器1045。

[0091] 处理器1170和1180被示为分别包括集成存储器控制器(IMC)单元1172和1182。处理器1170还包括作为其总线控制器单元的一部分的点对点(P-P)接口1176和1178;类似地,

第二处理器1180包括点对点接口1186和1188。处理器1170、1180可以使用点对点(P-P)接口电路1178、1188经由P-P接口1150来交换信息。如图11所示，IMC 1172和1182将各处理器耦合至相应的存储器，即存储器1132和存储器1134，这些存储器可以是本地附连至相应的处理器的主存储器的部分。

[0092] 处理器1170、1180可各自经由使用点对点接口电路1176、1194、1186、1198的各个P-P接口1152、1154与芯片组1190交换信息。芯片组1190可以可选地经由高性能接口1139与协处理器1138交换信息。在一个实施例中，协处理器1138是专用处理器，诸如例如高吞吐量MIC处理器、网络或通信处理器、压缩引擎、图形处理器、GPGPU、或嵌入式处理器等等。

[0093] 共享高速缓存(未示出)可以被包括在任一处理器之内，或被包括在两个处理器外部但仍经由P-P互连与这些处理器连接，从而如果将某处理器置于低功率模式时，可将任一处理器或两个处理器的本地高速缓存信息存储在该共享高速缓存中。

[0094] 芯片组1190可经由接口1196耦合至第一总线1116。在一个实施例中，第一总线1116可以是外围组件互连(PCI)总线，或诸如PCI Express总线或其它第三代I/O互连总线之类的总线，但本发明的范围并不受此限制。

[0095] 如图11所示，各种I/O设备1114可以连同总线桥1118耦合到第一总线1116，该总线桥将第一总线1116耦合至第二总线1120。在一个实施例中，诸如协处理器、高吞吐量MIC处理器、GPGPU的处理器、加速器(诸如例如图形加速器或数字信号处理(DSP)单元)、现场可编程门阵列或任何其它处理器的一个或多个附加处理器1115耦合到第一总线1116。在一个实施例中，第二总线1120可以是低引脚计数(LPC)总线。各种设备可以被耦合至第二总线1120，在一个实施例中这些设备包括例如键盘/鼠标1122、通信设备1127以及诸如可包括指令/代码和数据1130的盘驱动器或其它大容量存储设备的存储单元1128。此外，音频I/O 1124可以被耦合至第二总线1120。注意，其它架构是可能的。例如，代替图11的点对点架构，系统可以实现多分支总线或其它这类架构。

[0096] 现在参考图12，所示为根据本发明的实施例的更具体的第二示例性系统1200的框图。图11和图12中的相同部件用相同附图标记表示，并从图12中省去了图11中的某些方面，以避免使图12的其它方面变得模糊。

[0097] 图12示出处理器1170、1180可分别包括集成存储器和I/O控制逻辑("CL")1172和1182。因此，CL 1172、1182包括集成存储器控制器单元并包括I/O控制逻辑。图12不仅示出存储器1132、1134耦合至CL 1172、1182，而且还示出I/O设备1214也耦合至控制逻辑1172、1182。传统I/O设备1215被耦合至芯片组1190。

[0098] 现在参照图13，所示出的是根据本发明一个实施例的SoC 1300的框图。在图9中，相似的部件具有同样的附图标记。另外，虚线框是更先进的SoC的可选特征。在图13中，互连单元1302被耦合至：应用处理器1310，该应用处理器包括一个或多个核202A-N的集合以及共享高速缓存单元906；系统代理单元910；总线控制器单元916；集成存储器控制器单元914；一组或一个或多个协处理器1320，其可包括集成图形逻辑、图像处理器、音频处理器和视频处理器；静态随机存取存储器(SRAM)单元1330；直接存储器存取(DMA)单元1332；以及用于耦合至一个或多个外部显示器的显示单元1340。在一个实施例中，协处理器1320包括专用处理器，诸如例如网络或通信处理器、压缩引擎、GPGPU、高吞吐量MIC处理器、或嵌入式处理器等等。

[0099] 本文公开的机制的各实施例可以被实现在硬件、软件、固件或这些实现方法的组合中。本发明的实施例可实现为在可编程系统上执行的计算机程序或程序代码，该可编程系统包括至少一个处理器、存储系统(包括易失性和非易失性存储器和/或存储元件)、至少一个输入设备以及至少一个输出设备。

[0100] 可将程序代码(诸如图11中示出的代码1130)应用于输入指令，以执行本文描述的各功能并生成输出信息。可以按已知方式将输出信息应用于一个或多个输出设备。为了本申请的目的，处理系统包括具有诸如例如数字信号处理器(DSP)、微控制器、专用集成电路(ASIC)或微处理器之类的处理器的任何系统。

[0101] 程序代码可以用高级程序化语言或面向对象的编程语言来实现，以便与处理系统通信。在需要时，也可用汇编语言或机器语言来实现程序代码。事实上，本文中描述的机制不限于任何特定编程语言的范围。在任一情形下，该语言可以是编译语言或解释语言。

[0102] 至少一个实施例的一个或多个方面可以由存储在机器可读介质上的表示性指令来实现，指令表示处理器中的各种逻辑，指令在被机器读取时使得该机器制作用于执行本文所述的技术的逻辑。被称为“IP核”的这些表示可以被存储在有形的机器可读介质上，并被提供给多个客户或生产设施以加载到实际制造该逻辑或处理器的制造机器中。

[0103] 这样的机器可读存储介质可以包括但不限于通过机器或设备制造或形成的物品的非瞬态的有形安排，其包括存储介质，诸如：硬盘；任何其它类型的盘，包括软盘、光盘、紧致盘只读存储器(CD-ROM)、紧致盘可重写(CD-RW)以及磁光盘；半导体器件，例如只读存储器(ROM)、诸如动态随机存取存储器(DRAM)和静态随机存取存储器(SRAM)之类的随机存取存储器(RAM)、可擦除可编程只读存储器(EPROM)、闪存、电可擦除可编程只读存储器(EEPROM)；相变存储器(PCM)；磁卡或光卡；或适于存储电子指令的任何其它类型的介质。

[0104] 因此，本发明的各实施例还包括非瞬态的有形机器可读介质，该介质包含指令或包含设计数据，诸如硬件描述语言(HDL)，它定义本文中描述的结构、电路、装置、处理器和/或系统特征。这些实施例也被称为程序产品。

[0105] 虽然已经描述并在附图中示出了特定示例实施例，但可以理解，这些实施例仅仅是对本宽泛发明的说明而非限制，并且本发明不限于所示出和所描述的特定结构和配置，因为本领域技术人员在研究了本公开文本之后可以料知到多种其它修改方式。在诸如本申请这样的技术领域中，因为发展很快且未来的进步难以预见，所以本公开的诸个实施例可通过受益于技术进步而容易地获得配置和细节上的改动，而不背离本公开的原理和所附权利要求书的范围。

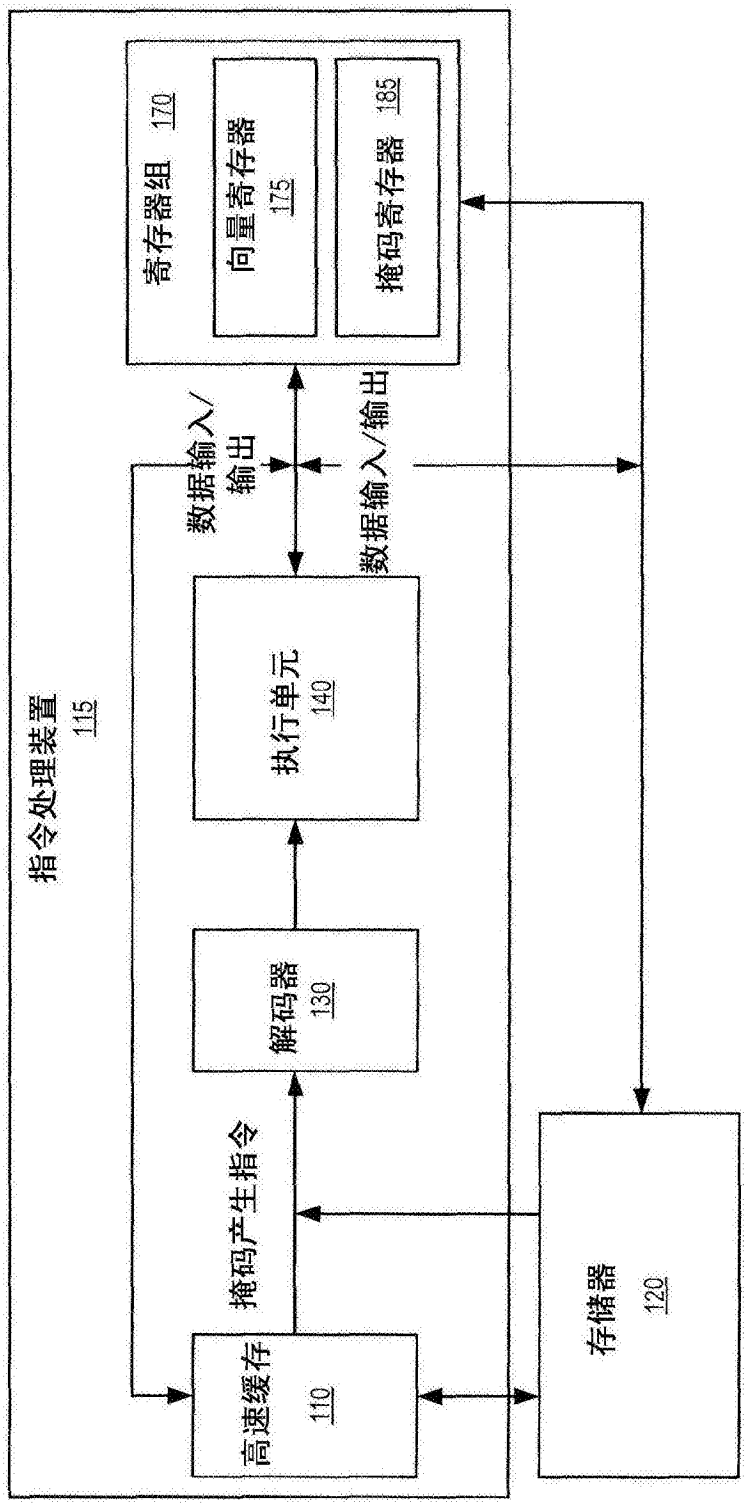

指令处理装置
115
寄存器组 170
向量寄存器
175
掩码寄存器 185
数据输入/
输出
数据输入/输出
执行单元
140
解码器
130
掩码产生指令
高速缓存
110
存储器
120

图1

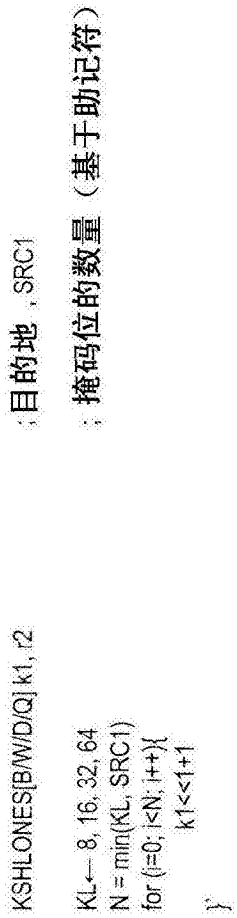

KSHLONES[B/W/D/Q] k1, r2
;目的地，SRC1
;掩码位的数量（基于助记符）
KL← 8, 16, 32, 64
N = min(KL, SRC1)
for (i=0; i<N; i++){
k1<<1+1
}

图2A

```
SHLONES[B/W/D/Q] r1, r2          ;目的地，SRC1

KL← 8, 16, 32, 64                ;掩码位的数量（基于助记符）
N = min(KL, SRC1)
for (i=0; i<N; i++){
        r1<<1+1
}
```

图2B

```
KSHLONES[B/W/D/Q] k1, r2        ; 目的地，SRC1

KL← 8, 16, 32, 64               ; 掩码位的数量（基于助记符）
N = min(KL, SRC1)
for (i=0; i<N; i++){
        k1<<1+1
}
if (N==0) ZF = 1
if (N==KL) CF = 1
```

图2C

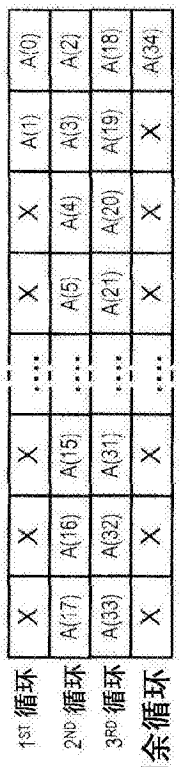

1ST 循环
2ND 循环
剩余循环
A(15) A(14) A(13) .... A(3) A(2) A(1) A(0)
A(31) A(30) A(29) .... A(19) A(18) A(17) A(16)
X X X .... X A(34) A(33) A(32)

图 3A

1ST 循环
2ND 循环
3RD 循环
剩余循环
X X X .... X A(5) A(4) A(3) A(2)
A(17) A(16) A(15) .... A(7) A(6) A(21) A(20) A(19) A(18)
A(33) A(32) A(31) .... A(23) A(22) X X X X
X X X .... X X X X X

图 3B

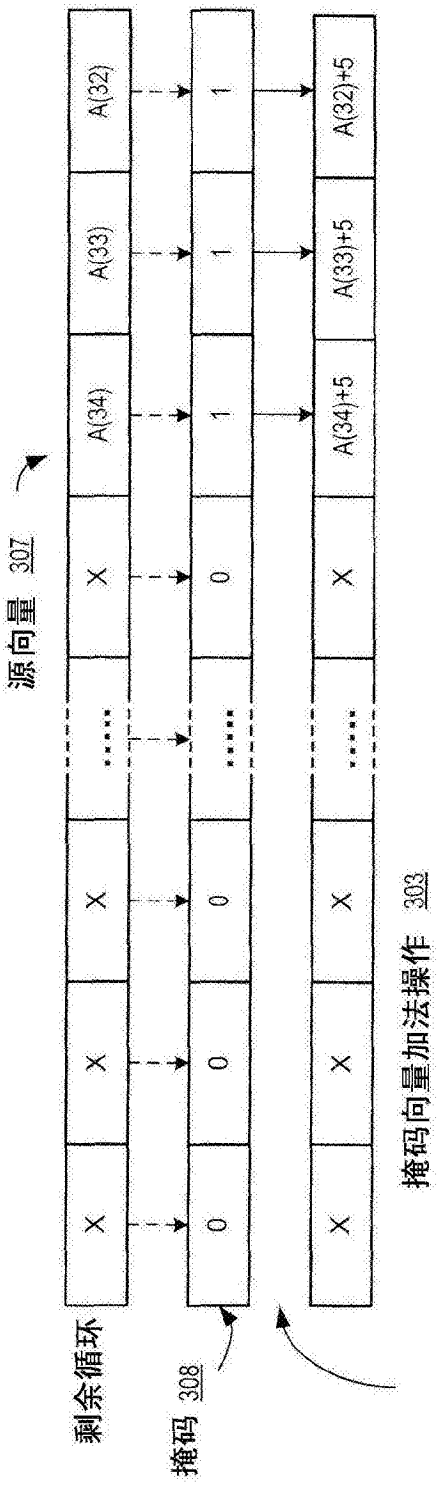

源向量 307
A(34)
A(33)
A(32)
X
.....
1
0
A(34)+5
A(33)+5
A(32)+5
剩余循环
掩码 308
掩码向量加法操作 303

图3C

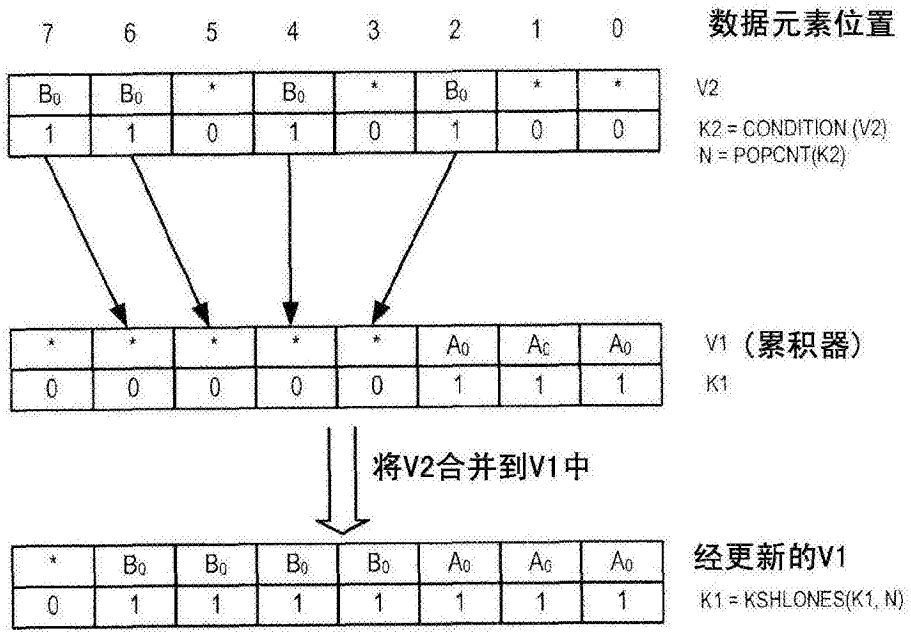

7 6 5 4 3 2 1 0
数据元素位置
B0 B0 * B0 * B0 * *
1 1 0 1 0 1 0 0
V2
K2 = CONDITION (V2)
N = POPCNT(K2)
* * * * * A0 A0 A0
0 0 0 0 0 1 1 1
V1（累积器）
K1
将V2合并到V1中
* B0 B0 B0 B0 A0 A0 A0
0 1 1 1 1 1 1 1
经更新的V1
K1 = KSHLONES(K1, N)

图4

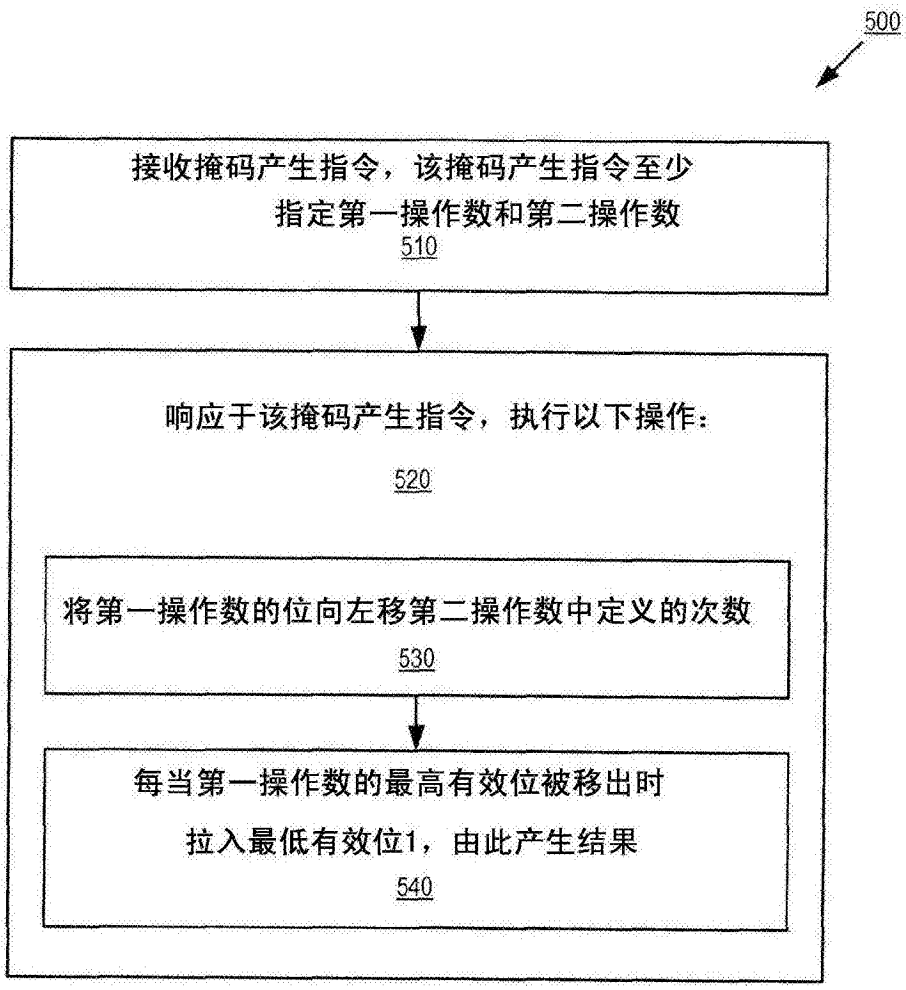

500
接收掩码产生指令，该掩码产生指令至少
指定第一操作数和第二操作数
510
响应于该掩码产生指令，执行以下操作：
520
将第一操作数的位向左移第二操作数中定义的次数
530
每当第一操作数的最高有效位被移出时
拉入最低有效位1，由此产生结果
540

图5

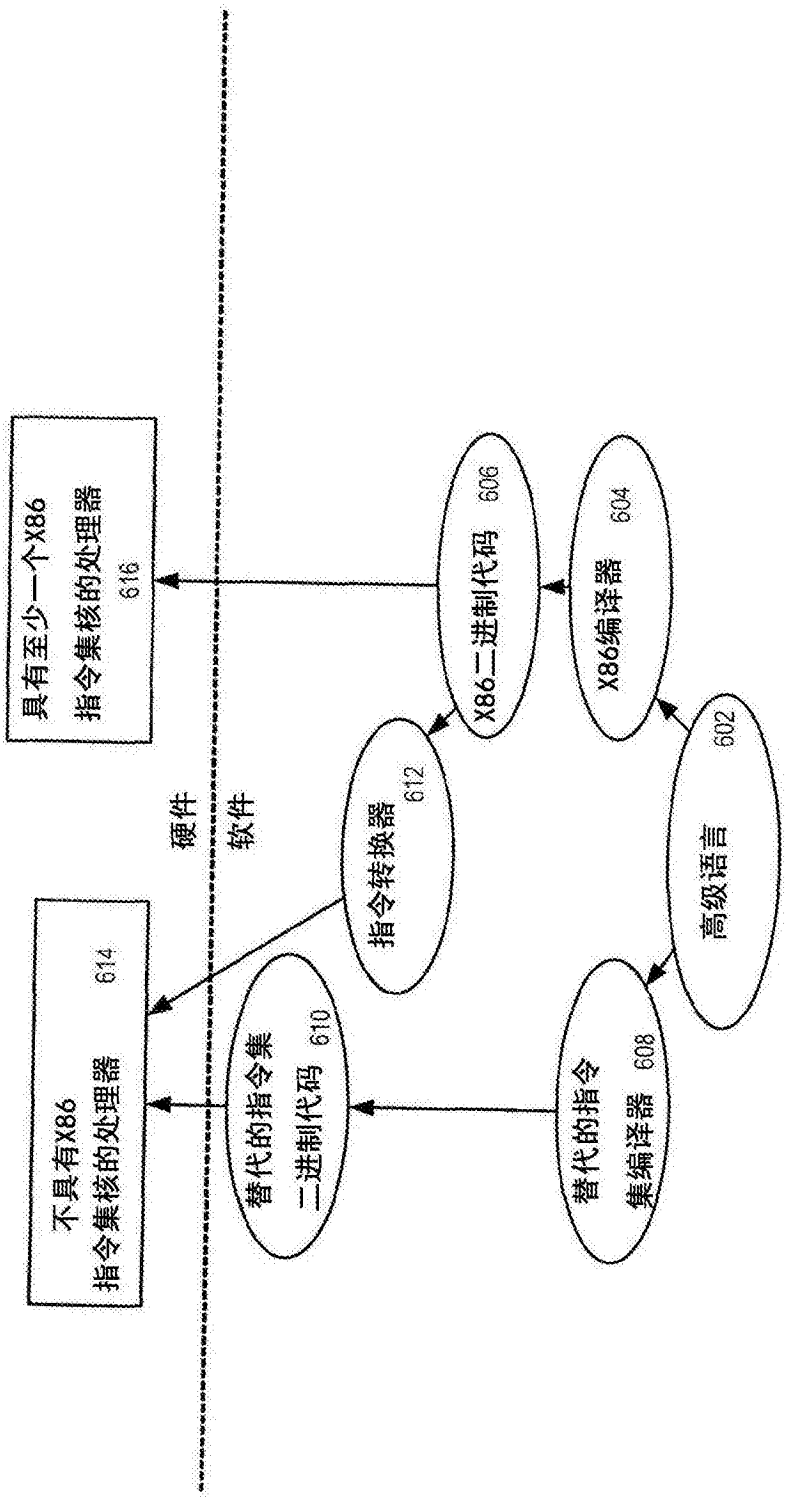

具有至少一个X86
指令集核的处理器
616
硬件
软件
X86二进制代码 606
X86编译器 604
指令转换器 612
高级语言 602
不具有X86
指令集核的处理器 614
替代的指令集
二进制代码 610
替代的指令
集编译器 608

图6

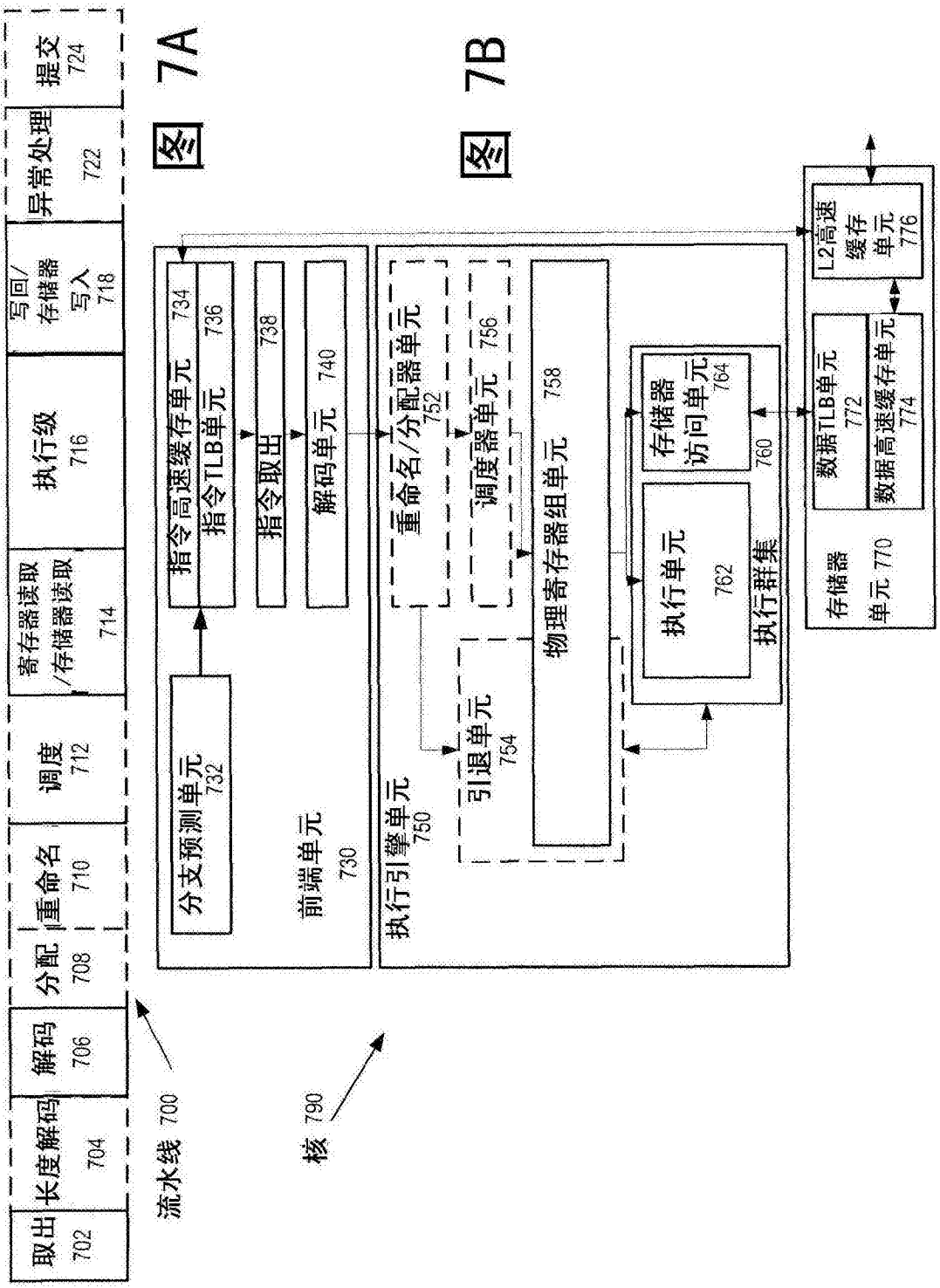

流水线 700
取出 702
长度解码 704
解码 706
分配 708
重命名 710
调度 712
寄存器读取/存储器读取 714
执行级 716
写回/存储器写入 718
异常处理 722
提交 724
前端单元 730
分支预测单元 732
指令高速缓存单元 734
指令TLB单元 736
指令取出 738
解码单元 740
执行引擎单元 750
重命名/分配器单元 752
引退单元 754
调度器单元 756
物理寄存器组单元 758
执行群集 760
执行单元 762
存储器访问单元 764
存储器单元 770
数据TLB单元 772
数据高速缓存单元 774
L2高速缓存单元 776
核 790

图 7A

图 7B

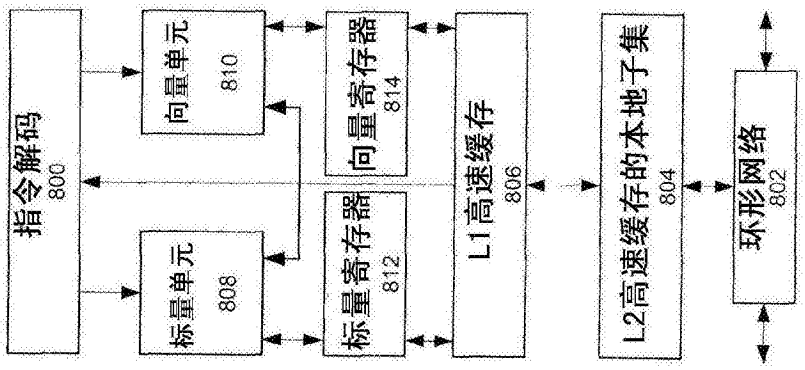

指令解码
800
向量单元
810
标量单元
808
向量寄存器
814
标量寄存器
812
L1高速缓存
806
L2高速缓存的本地子集
804
环形网络
802

图8A

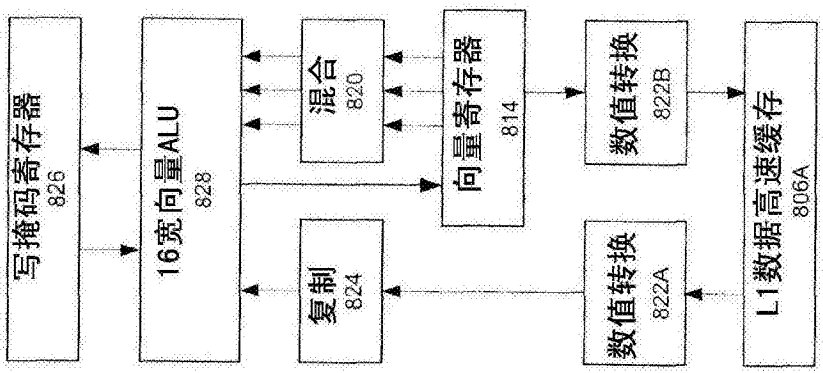

写掩码寄存器
826
16宽向量ALU
828
混合
820
复制
824
向量寄存器
814
数值转换
822B
数值转换
822A
L1数据高速缓存
806A

图8B

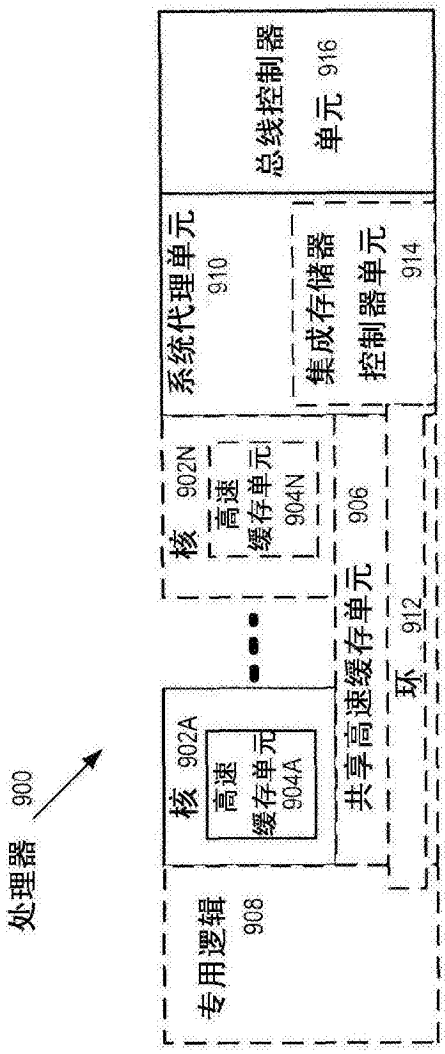

处理器
900
核
902A
高速
缓存单元
904A
核
902N
高速
缓存单元
904N
专用逻辑
908
共享高速缓存单元
906
环
912
系统代理单元
910
集成存储器
控制器单元
914
总线控制器
单元
916

图9

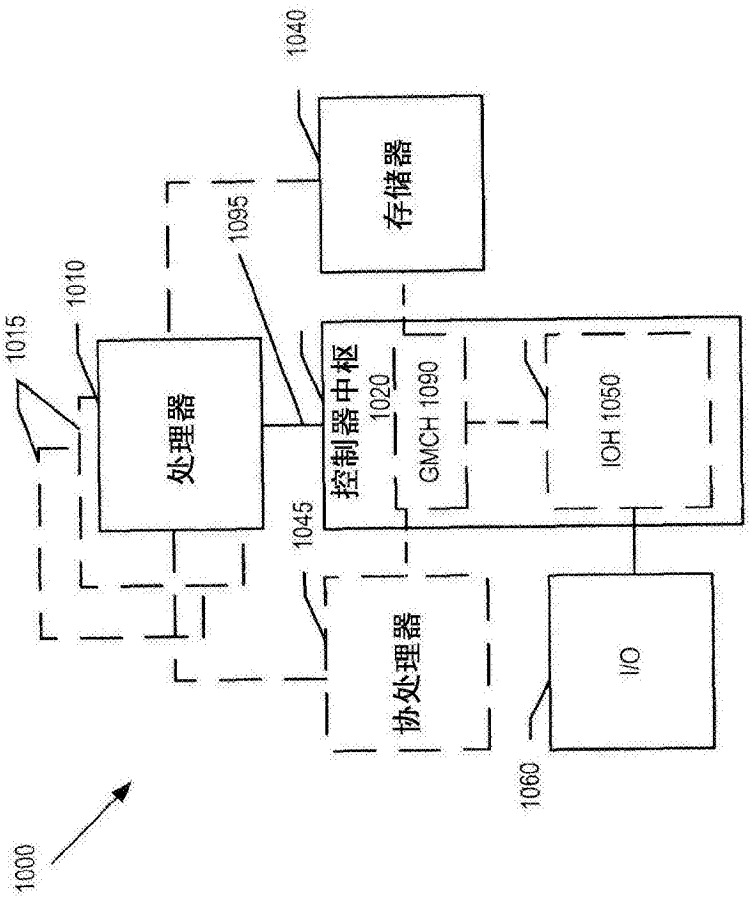

1040
存储器
1095
1010
1015
处理器
控制器中枢
1020
GMCH 1090
IOH 1050
1045
协处理器
I/O
1060
1000

图10

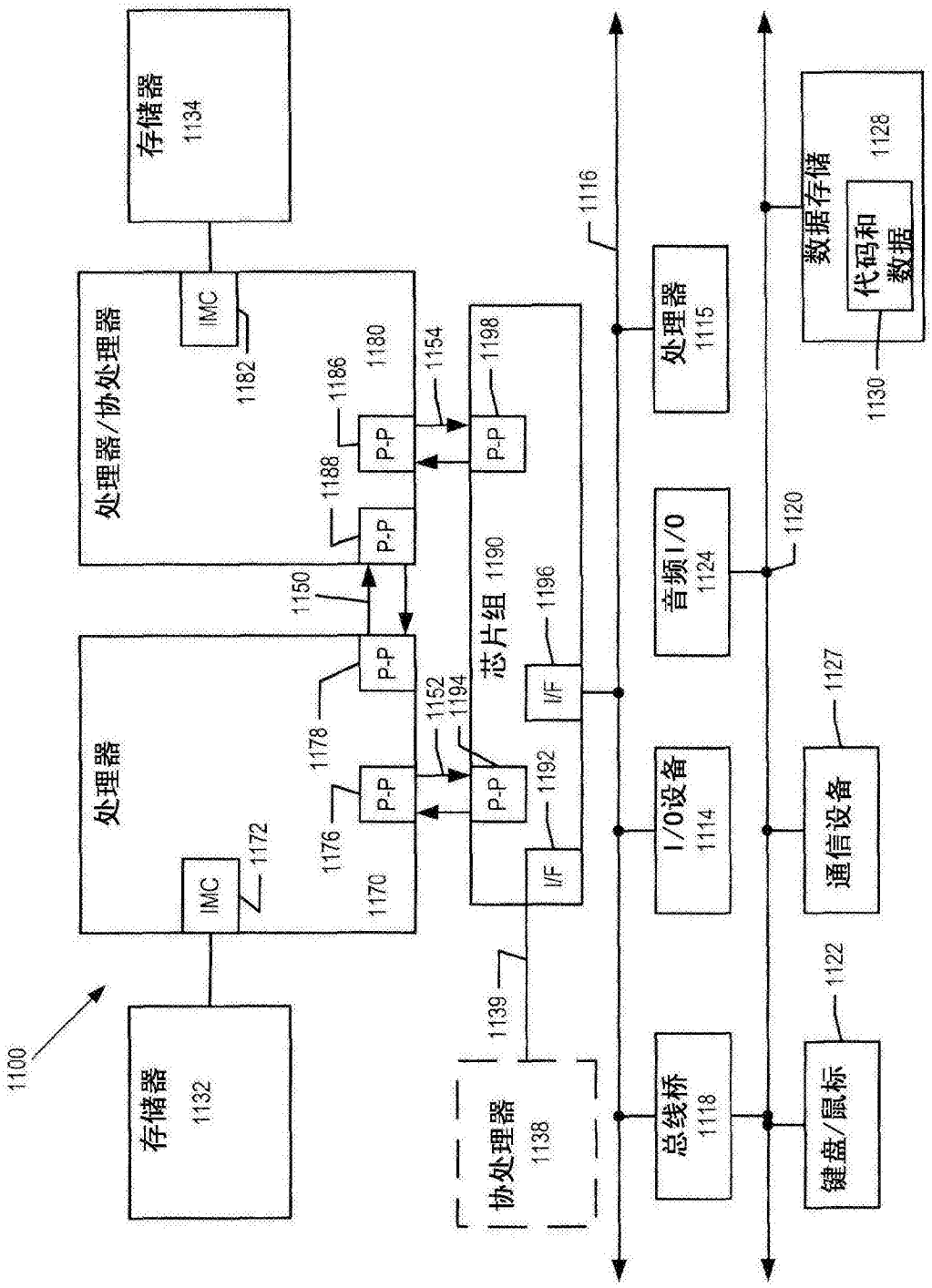

1100
存储器
1132
处理器
IMC
1172
1170
P-P
P-P
1176
1178
处理器/协处理器
IMC
1182
存储器
1134
P-P
P-P
1186
1188
1180
1154
1150
1152
1194
芯片组 1190
P-P
P-P
1198
1196
I/F
I/F
1192
1139
协处理器
1138
1116
处理器
1115
音频I/O
1124
1120
I/O设备
1114
通信设备
1127
总线桥
1118
键盘/鼠标
1122
数据存储
1128
代码和
数据
1130

图11

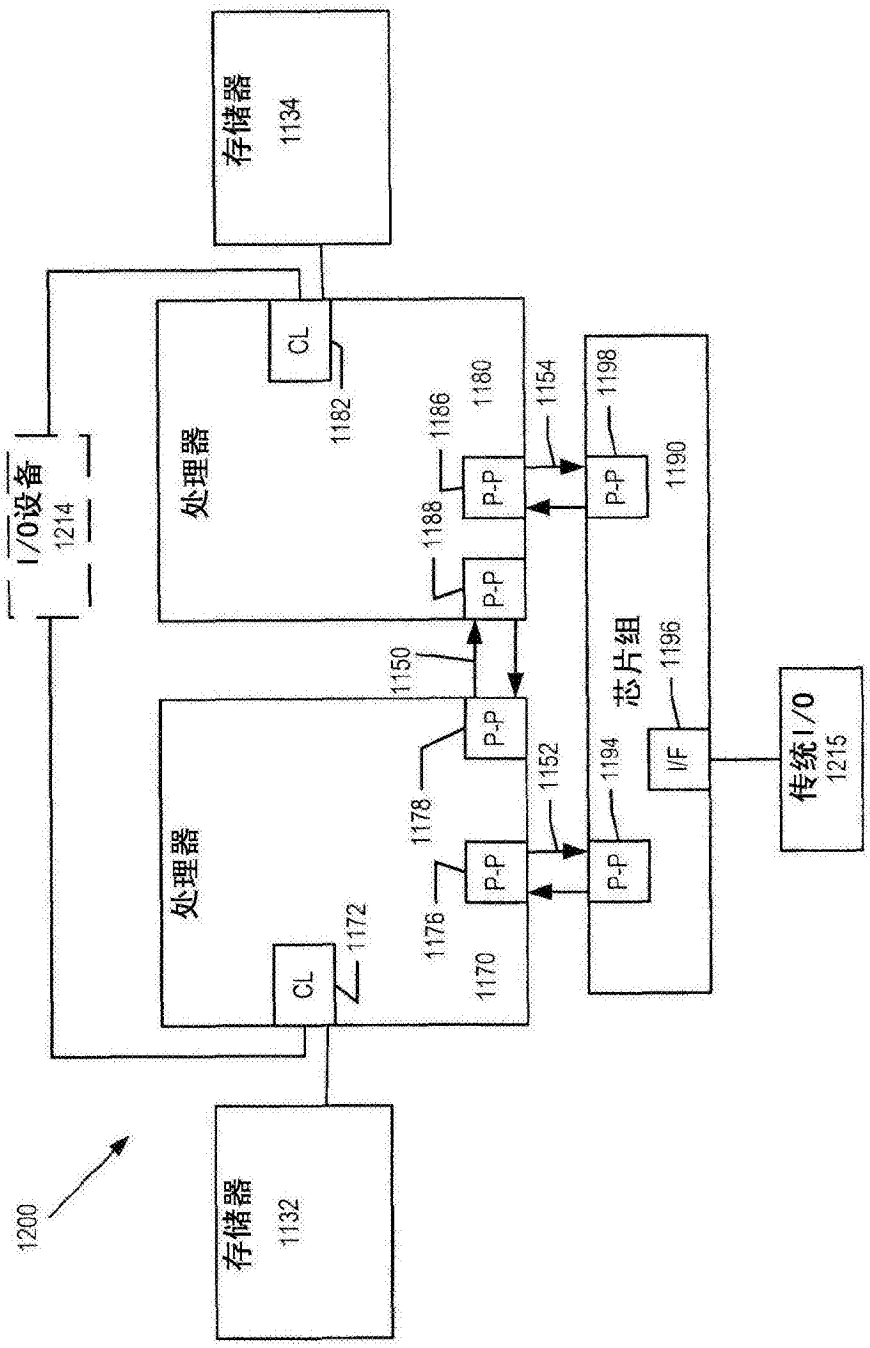

存储器
1134
处理器
CL
1182
1180
1186
P-P
1188
P-P
1154
1198
P-P
1190
芯片组
1196
I/F
传统I/O
1215
I/O设备
1214
1150
处理器
P-P
1178
1152
1194
P-P
P-P
1176
1170
CL
1172
1200
存储器
1132

图12

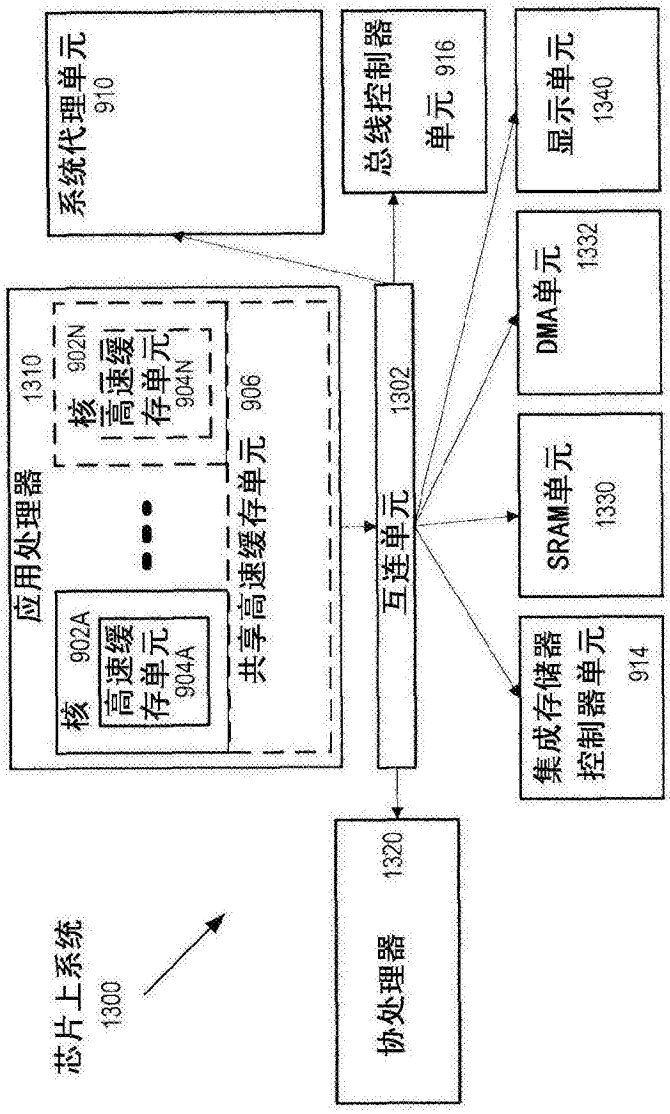

芯片上系统
1300
应用处理器
1310
核
902A
高速缓存单元
904A
核
902N
高速缓存单元
904N
共享高速缓存单元
906
互连单元
1302
协处理器
1320
系统代理单元
910
总线控制器单元
916
显示单元
1340
DMA单元
1332
SRAM单元
1330
集成存储器控制器单元
914

图13