(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 102103487 A
(43)申请公布日 2011.06.22

(21)申请号 201010601696.5

(22)申请日 2010.12.15

(30)优先权数据
12/653,704 2009.12.17 US

(71)申请人 英特尔公司
地址 美国加利福尼亚州

(72)发明人 V·戈帕尔 J·D·吉尔福德
E·奥兹图科 W·K·费格哈利
G·M·沃尔里齐 M·G·迪克森

(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 毛力

(51)Int.Cl.
G06F 9/305(2006.01)
G06F 9/315(2006.01)

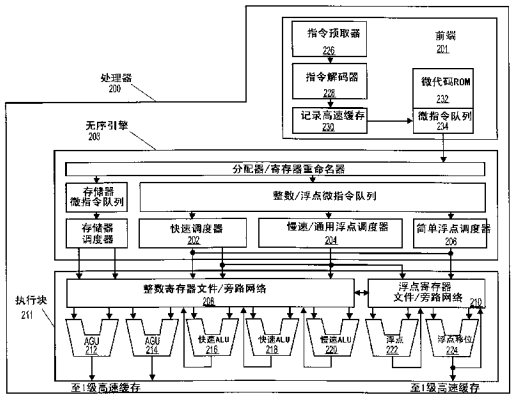
权利要求书 2 页 说明书 12 页 附图 8 页

(54)发明名称

用于在单个指令中执行移位和异或运算的方法和装置

(57)摘要

用于执行移位和异或运算的方法和装置。在一个实施例中,一种装置包括用于执行第一指令的执行资源。响应于该第一指令,所述执行资源对至少一个值执行移位和异或运算。



CN 102103487 A

1. 一种处理器，包括：

用于执行移位和异或指令的逻辑，其中将第一值移位移位量，且经移位值与第二值进行异或运算。

2. 如权利要求 1 所述的处理器，其特征在于，所述第一值将被左移位。

3. 如权利要求 1 所述的处理器，其特征在于，所述第一值将被右移位。

4. 如权利要求 1 所述的处理器，其特征在于，所述第一值通过逻辑方式移位。

5. 如权利要求 1 所述的处理器，其特征在于，所述第一值通过算术方式移位。

6. 如权利要求 1 所述的处理器，其特征在于，包括移位器和异或电路。

7. 如权利要求 1 所述的处理器，其特征在于，所述移位和异或指令包括用于存储所述第二值的第一字段。

8. 如权利要求 1 所述的处理器，其特征在于，所述第一值是压缩数据类型。

9. 一种系统，包括：

存储，用于存储用来执行移位和异或运算的第一指令；

处理器，用于执行用来执行移位和异或指令的逻辑，其中将第一值移位移位量，且经移位值与第二值进行异或运算。

10. 如权利要求 9 所述的系统，其特征在于，所述第一值将被左移位。

11. 如权利要求 9 所述的系统，其特征在于，所述第一值将被右移位。

12. 如权利要求 9 所述的系统，其特征在于，所述第一值通过逻辑方式移位。

13. 如权利要求 9 所述的系统，其特征在于，所述第一值通过算术方式移位。

14. 如权利要求 9 所述的系统，其特征在于，包括移位器和异或电路。

15. 如权利要求 9 所述的系统，其特征在于，所述移位和异或指令包括用于存储所述第二值的第一字段。

16. 如权利要求 9 所述的系统，其特征在于，所述第一值是压缩数据类型。

17. 一种方法，包括：

执行移位和异或指令，其中将第一值移位移位量，且经移位值与第二值进行异或运算。

18. 如权利要求 17 所述的方法，其特征在于，所述第一值将被左移位。

19. 如权利要求 17 所述的方法，其特征在于，所述第一值将被右移位。

20. 如权利要求 17 所述的方法，其特征在于，所述第一值通过逻辑方式移位。

21. 如权利要求 17 所述的方法，其特征在于，所述第一值通过算术方式移位。

22. 如权利要求 17 所述的方法，其特征在于，包括移位器和异或电路。

23. 如权利要求 17 所述的方法，其特征在于，所述移位和异或指令包括用于存储所述第二值的第一字段。

24. 如权利要求 17 所述的方法，其特征在于，所述第一值是压缩数据类型。

25. 一种其上存储有指令的机器可读介质，所述指令在由机器执行时使所述机器执行一种方法，所述方法包括：

将第一值移位移位量；以及

将经移位值与第二值进行异或运算。

26. 如权利要求 25 所述的机器可读介质，其特征在于，所述第一值将被左移位。

27. 如权利要求 25 所述的机器可读介质，其特征在于，所述第一值将被右移位。

28. 如权利要求 25 所述的机器可读介质，其特征在于，所述第一值通过逻辑方式移位。

29. 如权利要求 25 所述的机器可读介质，其特征在于，所述第一值通过算术方式移位。

30. 如权利要求 25 所述的机器可读介质，其特征在于，包括移位器和异或电路。

31. 如权利要求 25 所述的机器可读介质，其特征在于，所述移位和异或指令包括用于存储所述第二值的第一字段。

32. 如权利要求 25 所述的机器可读介质，其特征在于，所述第一值是压缩数据类型。

33. 一种方法，包括 ：

在第一经移位值与第二位折算值之间执行异或 (XOR) 运算，并将运算结果存储于第一寄存器中 ；

在所述结果中检查最少前导零数量。

34. 如权利要求 33 所述的方法，其特征在于，如果所述最少数量的前导零在所述结果中，则表明所述结果与第一组块相对应。

35. 如权利要求 34 所述的方法，其特征在于，所述第一经移位值将被左移 1 位。

36. 如权利要求 34 所述的方法，其特征在于，所述第一经移位值将被右移 1 位。

用于在单个指令中执行移位和异或运算的方法和装置

技术领域

[0001] 本发明属于计算机处理领域。更具体地，各个实施例涉及用于执行移位和异或(XOR)运算的指令。

背景技术

[0002] 单指令多数据(SIMD)指令在各种应用中可用于并行地处理多个数据元(压缩数据)。串行地执行诸如移位运算和异或(XOR)运算的运算会降低性能。

附图说明

[0003] 本发明通过示例进行说明，且不受限于附图的各个图，其中：

[0004] 图1A是根据本发明的一个实施例的计算机系统的框图，该计算机系统被形成为具有处理器，该处理器包括用于执行移位和异或运算指令的执行单元；

[0005] 图1B是根据本发明的替代实施例的另一示例性计算机系统的框图；

[0006] 图1C是根据本发明的另一替代实施例的又一示例性计算机系统的框图；

[0007] 图2是根据本发明的一个实施例的处理器的微体系结构的框图，该处理器包括用于执行移位和异或运算的逻辑电路；

[0008] 图3A示出根据本发明的一个实施例的多媒体寄存器中的各种压缩数据类型表示；

[0009] 图3B示出根据替代实施例的压缩数据类型；

[0010] 图3C示出根据本发明的一个实施例的多媒体寄存器中的各种有符号和无符号压缩数据类型表示；

[0011] 图3D示出运算编码(运算码)格式的一个实施例；

[0012] 图3E示出替代的运算编码(运算码)格式；

[0013] 图3F示出又一替代的运算编码格式；

[0014] 图4是根据本发明的用于执行指令的逻辑的一个实施例的框图。

[0015] 图5是要与一个实施例协同执行的运算的流程图。

具体实施方式

[0016] 以下说明书描述了一种用于在处理装置、计算机系统或软件程序中执行移位和异或运算的技术的实施例。在以下描述中，陈述了诸如处理器类型、微体系结构条件、事件、启用机制等来提供对本发明的更透彻理解。然而，本领域技术人员将理解，没有这些特定细节也可实施本发明。此外，未详细示出一些公知的结构、电路等，以避免不必要地混淆本发明的实施例。

[0017] 虽然参照处理器描述了以下实施例，但其他实施例也可应用于其他类型的集成电路和逻辑器件。本发明的相同技术和示教能容易地应用于可受益于较高流水线吞吐量和改进性能的其他类型的电路或半导体器件。本发明的示教可应用于执行数据操纵的任何处理

器或机器。然而，本发明的实施例不限于执行 256 位、128 位、64 位、32 位或 16 位数据运算的处理器或机器，且可应用于其中需要运算压缩数据的任何处理器和机器。

[0018] 虽然以下示例描述了执行单元和逻辑电路背景下的指令处理和分配，但本发明的其他实施例可通过存储于有形介质上的软件来实现。在一个实施例中，本发明的方法以机器可执行指令的方式具体化。这些指令可被用于使利用这些指令编程的通用或专用处理器执行本发明的步骤。本发明的多个实施例可作为计算机程序产品或软件提供，其可包括其上存储了计算机程序指令的计算机可读介质，这些计算机程序指令可用于对计算机（或其它电子设备）编程以执行根据本发明的过程。替代地，本发明的步骤可通过包含用于执行这些步骤的硬接线逻辑的特定硬件部件来执行，或通过已编程计算机部件和定制硬件部件的任何组合来执行。此类软件可被存储在系统中的存储器中。类似地，代码可经由网络分发，或通过其他计算机可读介质来分发。

[0019] 因此，计算机可读介质可包括用于以机器（计算机）可读的形式存储或发送信息的任何机构，包括但不限于软盘、光盘、光盘只读存储器 (CD-ROM)、以及磁光盘、只读存储器 (ROM)、随机存取存储器 (RAM)、可擦除可编程只读存储器 (EPROM)、电可擦除可编程只读存储器 (EEPROM)、磁或光卡、闪存、经由因特网的传输、电形式、光形式、声音形式或其他形式的传播信号（例如载波、红外信号、数字信号等）等。因此，计算机可读介质包括适用于以机器（例如计算机）可读的形式存储或发送电子指令或信息的任何类型的介质 / 机器可读介质。此外，本发明也可被下载为计算机程序产品。同样，程序可从远程计算机（例如服务器）被传输至请求计算机（例如客户机）。该程序的传输可通过被具体化在经由通信链路（例如调制解调器、网络连接等）的载波或其它传输介质中的电、光、声音或其它形式的数据信号的方式。

[0020] 一种设计可经过多个阶段，从创建到模拟到制造。表示设计的数据可按照多种方式来表示该设计。首先，因为在模拟中有用，该硬件可使用硬件描述语言或另一功能描述语言来表示。此外，可在设计过程的一些阶段制造具有逻辑和 / 或晶体管门电路的电路级模型。再者，在一些阶段，大多数设计达到表示各种器件在硬件模型中的物理布置的数据级。在使用常规半导体制造技术的情况下，表示硬件模型的数据可以是指定各种功能部件在用于生产集成电路的掩模的不同掩模层上存在或不存在的数据。在该设计的任何表示中，该数据可以机器可读介质的任何形式来存储。经调制或其它方式生成的用于发送此类信息的光或电波、存储器、诸如盘的磁或光存储可以是该机器可读介质。这些介质中的任一种可“承载”或“表明”该设计或软件信息。当表明或承载该代码或设计的电载波被发送时，只要执行该电信号的复制、缓存或重新发送，就建立了新的副本。因此，通信提供商或网络提供商可建立具体化本发明技术的物品（载波）的副本。

[0021] 在现代处理器中，多个不同执行单元被用于处理和执行各种代码和指令。并非所有指令都被等同地创建，因为一些指令较快地完成，而另一些会花费数目庞大的时钟周期。指令的吞吐越快，处理器的总体性能越好。因此，使许多指令尽可能快地执行会是有利的。然而，存在具有更高复杂度且在执行时间和处理器资源方面需要更多的某些指令。例如，存在浮点指令、装载 / 存储操作、数据移动等。

[0022] 随着越来越多的计算机系统被用于因特网和多媒体应用，已随时间引入了附加的处理器支持。例如，单指令多数据 (SIMD) 整数 / 浮点指令和流送 SIMD 扩展 (SSE) 是减少

执行特定程序任务所需的总指令数的指令，这又减少了功耗。这些指令可通过对多个数据元进行并行运算来加速软件性能。因此，可在包括视频、语音以及图像 / 照片处理的宽泛范围的应用中实现性能提高。SIMD 指令在微处理器和相似类型的逻辑电路中的实现通常涉及多种问题。此外，SIMD 运算的复杂度通常导致对用于正确处理和操纵数据的附加电路的需要。

[0023] 当前，尚无 SIMD 移位和异或指令可用。在不存在 SIMD 移位和异或指令的情况下，根据本发明的实施例，可能需要大量指令和数据寄存器来在诸如音频 / 视频 / 图形压缩、处理以及操纵的应用中获得相同结果。因此，根据本发明的实施例的至少一个移位和异或指令可减少代码开销和资源要求。本发明的实施例提供利用 SIMD 相关硬件将移位和异或运算实现为算法的方法。当前，对 SIMD 寄存器中的数据执行移位和异或运算是有些困难和冗长的。相比于执行那些运算的实际指令数量，一些算法需要更多指令用于安排数据进行算术运算。通过实现根据本发明的实施例的移位和异或运算的实施例，实现移位和异或处理所需的指令的数量可显著减少。

[0024] 本发明的实施例涉及用于实现移位和异或运算的指令。根据一个实施例对数据元进行的移位和异或运算可一般表示如下：

[0025] DEST1 ← SRC1[SRC2]；

[0026] 在一个实施例中，SRC1 存储具有多个数据元的第一运算数，而 SRC2 包含表示要被该移位和异或指令移动多少值的值。在其它实施例中，移位和异或值指示符可被存储于紧邻的字段中。

[0027] 在上述流程中，“DEST”（目地的）和“SRC”（源）是用于表示相应数据或运算的源和目的地的通用术语。在一些实施例中，它们可通过寄存器、存储器或具有与所描述的名称或功能不同的其它存储区来实现。例如，在一个实施例中，DEST1 和 DEST2 可以是第一和第二临时存储区（例如“TEMP1”和“TEMP2”寄存器），SRC1 和 SRC3 可以是第一和第二目的地存储区（例如“DEST1”和“DEST2”寄存器），如此等等。在其它实施例中，SRC 和 DEST 存储区中的两个或更多个可与同一存储区（例如 SIMD 寄存器）内的不同数据存储元相对应。

[0028] 图 1A 是根据本发明一个实施例的示例性计算机系统的框图，该计算机系统被形成为具有处理器，该处理器包括用于执行移位和异或运算的指令的执行单元。根据本发明，系统 100 包括诸如处理器 102 的部件，该部件采用包括执行诸如本文所描述的实施例中的用于处理数据的算法的逻辑。系统 100 代表基于可从美国加利福尼亚州圣克拉拉市的英特尔公司买到的PENTIUM® III、PENTIUM® 4、Xeon™、Itanium®、XScale™ 和 / 或 StrongARM™ 微处理器的处理系统，不过也可使用其它系统（包括具有其它微处理器的 PC、工程工作站、机顶盒等）。在一个实施例中，样本系统 100 可执行可从美国华盛顿州雷蒙德市的微软公司买到的 WINDOWS™ 操作系统的一个版本，不过也可使用其它操作系统（例如 UNIX 和 Linux）、嵌入式软件、和 / 或图形用户界面。因此，本发明的实施例不限于硬件电路和软件的任何特定组合。

[0029] 诸实施例不限于计算机系统。本发明的替代实施例也可用于诸如手持设备和嵌入式应用的其它设备。手持设备的一些示例包括蜂窝电话、网际协议设备、数码相机、个人数字助理 (PDA) 以及手持 PC。嵌入式应用可包括微控制器、数字信号处理器 (DSP)、芯片上系统、网络计算机（网络 PC）、机顶盒、网络集线器、广域网 (WAN) 交换机或对操作数执行移位

和异或运算的任何其它系统。此外，已实现一些体系结构以使指令能同时运算若干数据，以提高多媒体应用的效率。随着数据类型和数据量增加，计算机和它们的处理器必须被增强以按照更高效的方法操纵数据。

[0030] 图 1A 是根据本发明一个实施例的计算机系统 100 的框图，该计算机系统 100 被形成为具有处理器 102，该处理器 102 包括用于执行对多个数据元的移位和异或算法的一个或更多个执行单元 108。一个实施例可在单处理器桌面或服务器系统的背景下进行描述，但替代实施例可被包括在多处理器系统中。系统 100 是中枢体系结构的示例。计算机系统 100 包括用于处理数据信号的处理器 102。处理器 102 可以是复杂指令集计算机 (CISC) 微处理器、简约指令集计算 (RISC) 微处理器、超长指令字 (VLIW) 微处理器、实现指令集组合的处理器或诸如例如数字信号处理器的任何其他处理器设备。处理器 102 耦合至处理器总线 110，该处理器总线 110 可在处理器 102 与系统 100 中的其它部件之间传输数据信号。系统 100 的元件执行本领域技术人员熟知的它们的常规功能。

[0031] 在一个实施例中，处理器 102 包括一级 (L1) 内部高速缓存 104。取决于体系结构，处理器 102 可具有单级内部高速缓存或多级内部高速缓存。替代地，在另一实施例中，高速缓存可驻留在处理器 102 外部。取决于特定实现和需要，其它实施例也可包括内部和外部高速缓存的组合。寄存器文件 106 可在包括整数寄存器、浮点寄存器、状态寄存器以及指令指针寄存器的各种寄存器中存储不同类型的数据。

[0032] 包括用于执行整数和浮点运算的逻辑的执行单元 108 也可驻留于处理器 102 中。处理器 102 还包括存储用于某些宏指令的微代码的微代码 (ucode)ROM。对于该实施例而言，执行单元 108 包括用于处理压缩指令集 109 的逻辑。在一个实施例中，压缩指令集 109 包括用于对多个运算数执行移位和异或的压缩移位和异或指令。通过将压缩指令集 109 包括在通用处理器 102 以及用于执行指令的关联电路的指令集中，许多多媒体应用所使用的操作可使用通用处理器 102 中的压缩数据来执行。因此，通过将处理器数据总线的全带宽用于对压缩数据执行运算，可使许多多媒体应用加速和更高效地执行。这样可消除在处理器数据总线上传输较小单位数据以在一个时刻对数据元执行一个或更多个运算的需要。

[0033] 执行单元 108 的替代实施例也可在微控制器、嵌入式处理器、图形设备、DSP 以及其它类型的逻辑电路中使用。系统 100 包括存储器 120。存储器 120 可以是动态随机存取存储器 (DRAM) 器件、静态随机存取存储器 (SRAM) 器件、闪存器件或其它存储器件。存储器 120 可存储通过数据信号表示的可由处理器 102 执行的指令和 / 或数据。

[0034] 系统逻辑芯片 116 耦合至处理器总线 110 和存储器 120。所示实施例中的系统逻辑芯片 116 是存储器控制中枢 (MCH)。处理器 102 可经由处理器总线 110 向 MCH 116 通信。MCH 116 向存储器 120 提供高带宽存储器通道 118，以用于存储指令和数据且用于存储图形命令、数据和纹理。MCH116 用于在处理器 102、存储器 120 以及系统 100 中的其它部件之间引导数据信号，并在处理器总线 110、存储器 120 以及系统 I/O 122 之间桥接数据信号。在一些实施例中，系统逻辑芯片 116 可提供用于耦合至图形控制器 112 的图形端口。MCH 116 通过存储器接口 118 耦合至存储器 120。图形卡 112 通过加速图形端口 (AGP) 互连 114 耦合至 MCH 116。

[0035] 系统 100 使用专用中枢接口总线 122 将 MCH 116 耦合至 I/O 控制器中枢 (ICH)130。ICH 130 经由本地 I/O 总线向一些 I/O 设备提供直接连接。本地 I/O 总线是用于将外围设

备连接至存储器 120、芯片组以及处理器 102 的高速 I/O 总线。一些示例是音频控制器、固件集线器（闪存 BIOS）128、无线收发器 126、数据存储 124、包含用户输入和键盘接口的传统 I/O 控制器、诸如通用串行总线 (USB) 的串行扩展端口以及网络控制器 134。数据存储设备 124 可包括硬盘驱动器、软盘驱动器、CD-ROM 设备、闪存器件或其它大容量存储设备。

[0036] 对于系统的另一实施例而言，用于执行具有移位和异或指令的算法的执行单元可与芯片上系统一起使用。芯片上系统的一个实施例由处理器和存储器组成。用于一种此类系统的存储器是闪存。该闪存可位于与处理器和其它系统部件相同的同一管芯上。此外，诸如存储器控制器或图形控制器的其它逻辑块也可位于芯片上系统上。

[0037] 图 1B 示出实现根据本发明一个实施例的原理的数据处理系统 140。本领域技术人员将容易理解，本文所描述的实施例可与替代的处理系统一起使用，而不背离本发明的范围。

[0038] 计算机系统 140 包括能执行包括移位和异或运算的 SIMD 操作的处理核 159。对于一个实施例而言，处理核 150 表示具有任何类型的体系结构的处理单元，包括但不限于 CISC、RISC 或 VLIW 型体系结构。处理核 159 也可适用于以一种或更多种工艺技术来制造，而且通过以足够的细节被表示在机器可读介质上，处理核 159 可适合便于所述制造。

[0039] 处理核 159 包括执行单元 142、一组寄存器文件 145 以及解码器 144。处理核 159 还包括对理解本发明而言不必要的附加电路（未示出）。执行单元 142 用于执行由处理核 159 接收的指令。除了识别典型的处理器指令之外，执行单元 142 可识别用于对压缩数据格式进行运算的压缩指令集 143 中的指令。压缩指令集 143 包括用于支持移位和异或运算的指令，且可包括其它压缩指令。执行单元 142 通过内部总线耦合至寄存器文件 145。寄存器文件 145 表示处理核 159 上的用于存储包括数据的信息的存储区。如上所述，应理解用于存储压缩数据的存储区不是关键的。执行单元 142 耦合至解码器 144。解码器 144 用于将处理核 159 所接收的指令解码成控制信号和 / 或微代码入口点。响应于这些控制信号和 / 或微代码入口点，执行单元 142 执行适当的操作。

[0040] 处理核 159 与总线 141 耦合以与各个其它系统设备通信，其它系统设备可包括但不限于，例如同步动态随机存取存储器 (SDRAM) 控制器 146、静态随机存取存储器 (SRAM) 控制器 147、猝发闪存接口 148、个人计算机存储卡国际协会 (PCMCIA)/ 紧致闪存 (CF) 卡控制器 149、液晶显示器 (LCD) 控制器 150、直接存储器存取 (DMA) 控制器 151 以及替代的总线主接口 152。在一个实施例中，数据处理系统 140 还可包括经由 I/O 总线 153 与各个 I/O 设备通信的 I/O 桥 154。此类 I/O 设备可包括但不限于例如统一异步接收器 / 发射器 (UART)155、通用串行总线 (USB)156、蓝牙无线 UART 157 和 I/O 扩展接口 158。

[0041] 数据处理系统 140 的一个实施例供移动、网络和 / 或无线通信之用，以及处理核 159 能执行包括移位和异或运算的 SIMD 操作。处理核 159 可用各种音频、视频、成像以及通信算法编程，包括 ：诸如沃尔什 - 哈达马德 (Walsh-Hadamard) 变换、快速傅立叶变换 (FFT)、离散正弦变换 (DCT) 的离散变换以及它们各自的逆变换 ；诸如色空间变换、视频编码运动估计或视频解码运动补偿的压缩 / 解压缩技术 ；以及诸如脉冲编码调制 (PCM) 的调制 / 解调 (MODEM) 功能。本发明的一些实施例也可应用于图形应用，诸如三维（“3D”）建模、渲染、对象碰撞检测、3D 对象变换和照明等。

[0042] 图 1C 示出能执行 SIMD 移位和异或运算的数据处理系统的另一实施例。根据一

个替代实施例，数据处理系统 160 可包括主处理器 166、SIMD 协处理器 161、高速缓存 167 以及输入 / 输出系统 168。输入 / 输出系统 168 可任选地耦合至无线接口 169。SIMD 协处理器 161 能执行包括移位和异或运算的 SIMD 运算。处理核 170 可适于以一种或更多种工艺技术来制造，且通过以足够的细节被表示在机器可读介质上，可适合便于包括处理核 170 的数据处理系统 160 的全部或一部分的制造。

[0043] 对于一个实施例而言，SIMD 协处理器 161 包括执行单元 162 和一组寄存器文件 164。主处理器 165 的一个实施例包括解码器 165，该解码器 165 用于识别指令集 163 中的指令，包括由执行单元 162 执行的 SIMD 移位和异或计算指令。对于替代实施例而言，SIMD 协处理器 161 也包括用于解码指令集 163 中的指令的解码器的至少一部分 165B。处理核 170 还包括对理解本发明的实施例而言不必要的附加电路（未示出）。

[0044] 在操作时，主处理器 166 执行数据处理指令流，该数据处理指令流控制一般类型的数据处理操作，包括与高速缓存 167 以及输入 / 输出系统 168 的交互。数据处理指令流内嵌有 SIMD 协处理器指令。主处理器 166 的解码器 165 将这些 SIMD 协处理器指令识别为应当由附连的 SIMD 协处理器 161 执行的类型。因此，主处理器 166 将这些 SIMD 协处理器指令（或表示 SIMD 协处理器指令的控制信号）发送到协处理器总线 166 上，在该总线 166 处它们被任何附连的 SIMD 协处理器接收。在该情况下，SIMD 协处理器 161 将接受并执行发送给它的任何已接收 SIMD 协处理器指令。

[0045] 数据可经由无线接口 169 接收以通过 SIMD 协处理器指令进行处理。作为一个示例，可以数字信号的形式接收语音通信，该数字信号可由 SIMD 协处理器指令处理，以再生代表该语音通信的数字音频样本。作为另一示例，已压缩音频和 / 或视频可以数字比特流的形式接收，该数字比特流可由 SIMD 协处理器指令处理，以再生数字音频样本和 / 或运动视频帧。对于处理核 170 的一个实施例而言，主处理器 166 以及 SIMD 协处理器 161 被集成到单个处理核 170 中，该处理核 170 包括执行单元 162、一组寄存器文件 164 以及用于识别包括 SIMD 移位和异或指令的指令集 163 中的指令的解码器 165。

[0046] 图 2 是根据本发明一个实施例的包括用于执行移位和异或指令的逻辑电路的处理器 200 的微体系结构的框图。对于移位和异或指令的一个实施例而言，该指令可将浮点尾数值向右移动由指数所表示的量，将经移位值与一个值进行异或运算，并产生最终结果。在一个实施例中，有序前端 201 是处理器 200 的一部分，其读取要执行的宏指令，并将它们准备就绪以便于后来在处理器流水线中使用。该前端 201 可包括若干单元。在一个实施例中，指令预取器 226 从存储器读取宏指令，并将它们馈送到指令解码器 228 中，指令解码器 228 又将它们解码成机器可执行的称为微指令或微运算（也称为微 op 或 uop）的原语。在一个实施例中，追踪高速缓存 (trace cache)230 获取已解码的微指令，并将它们组装成微指令队列 234 中的程序有序序列或记录以供执行。当追踪高速缓存 230 遇到复杂宏指令时，微代码 ROM232 提供完成该运算所需的微指令。

[0047] 许多宏指令被转换成单个微指令，而其它宏指令需要若干微指令来完成整个操作。在一个实施例中，如果需要四个以上微指令来完成宏指令，则解码器 228 访问微代码 ROM 232 来执行该宏指令。对于一个实施例而言，压缩移位和异或指令可被解码成少量微指令以供在指令解码器 228 处进行处理。在另一实施例中，如果需要多个微指令来完成该运算，则用于压缩移位和异或算法的指令可被存储在微代码 ROM 232 中。追踪高速缓存 230

指的是用于确定在微代码 ROM 232 中读取用于移位和异或算法的微代码序列的正确微指令指针的入口点可编程逻辑阵列 (PLA)。在微代码 ROM 232 完成当前宏指令的微运算排序之后，机器的前端 201 恢复从追踪高速缓存 230 读取微运算。

[0048] 一些 SIMD 和其它多媒体类型的指令被认为是复杂指令。许多浮点相关指令也是复杂指令。因而，当指令解码器 228 遇到复杂宏指令时，在适当的位置访问微代码 ROM 232 以取回用于该宏指令的微代码序列。用于执行该宏指令的各个微运算被传递至无序执行引擎 203 以在适当的整数和浮点执行单元处执行。

[0049] 无序执行引擎 203 是微指令被准备好以供执行的地方。无序执行逻辑具有多个缓冲器，当微指令沿流水线向下并被排定以供执行时，这些缓冲器用于使微指令流平滑并对其重新排序以最优化性能。分配器逻辑分配每个微指令执行所需的机器缓冲器和资源。寄存器重命名逻辑将逻辑寄存器重命名到寄存器文件中的条目上。该分配器还为两个微指令队列（一个用于存储器操作，另一个用于非存储器操作）之一中的每个微指令分配条目，该分配器在以下指令调度器之前：存储器调度器、快速调度器 202、慢速 / 通用浮点调度器 204 以及简单浮点调度器 206。微指令调度器 202、204、206 基于微指令的相关输入寄存器操作数源的就绪和微指令完成它们的操作所需的执行资源的可用性来确定微指令何时准备好执行。本实施例的快速调度器 202 可在主时钟周期的每半个周期时调度，而其它调度器在每个主处理器时钟周期仅调度一次。调度器仲裁分派端口来排定微指令以供执行。

[0050] 寄存器文件 208、210 位于调度器 202、204、206 与执行块 211 中的执行单元 212、214、216、218、220、222、224 之间。存在分别用于整数和浮点运算的不同的寄存器文件 208、210。本实施例的每个寄存器文件 208、210 还包括旁路网络，该旁路网络可将尚未写入寄存器文件的刚完成的结果通过旁路发送或转发至新的相关微指令。整数寄存器文件 208 和浮点寄存器文件 210 也能相互传递数据。对于一个实施例，整数寄存器文件 208 被分割成两个独立的寄存器文件，一个寄存器文件用于数据的低位 32 位，而第二寄存器文件用于数据的高位 32 位。一个实施例的浮点寄存器文件 210 具有 128 位宽条目，因为浮点指令通常具有从 64 到 128 位宽的运算数。

[0051] 执行块 211 包含执行单元 212、214、216、218、220、222、224，指令实际在这些执行单元中执行。该部分包括存储微指令需要执行的整数和浮点数据运算数值的寄存器文件 208、210。本实施例的处理器 200 由多个执行单元组成：地址产生单元 (AGU)212、AGU 214、快速 ALU（算术逻辑单元）216、快速 ALU 218、慢速 ALU 220、浮点 ALU 222、浮点移动单元 224。对于该实施例，浮点执行块 222、224 执行浮点、MMX、SIMD 以及 SSE 运算。本实施例的浮点 ALU 222 包括用于执行除法、平方根以及其它微运算的 64 位 ×64 位浮点除法器。对于本发明的实施例而言，涉及浮点值的任何动作利用该浮点硬件进行。例如，整数格式和浮点格式之间的转换涉及浮点寄存器文件。类似地，浮点除法运算在浮点除法器处进行。另一方面，非浮点数和整数类型利用整数硬件资源进行处理。简单而非常频繁的 ALU 操作由高速 ALU 执行单元 216、218 完成。本实施例的快速 ALU 216、218 可在半个时钟周期的有效等待时间内执行快速运算。对于一个实施例而言，多数复杂整数运算由慢速 ALU 220 完成，因为慢速 ALU 220 包括用于长等待时间类型的运算的整数执行硬件，诸如乘法器、移位器、标记逻辑以及分支处理。存储器装载 / 存储操作由 AGU 212、214 执行。对于该实施例，整数 ALU 216、218、220 在对 64 位数据运算数执行整数运算的背景下进行描述。在替代实施例

中，可实现 ALU 216、218、220 以支持包括 16、32、128、256 的各种数据位。类似地，可实现浮点单元 222、224 以支持具有各种位宽的多个运算数。对于一个实施例而言，浮点单元 222、224 可协同 SIMD 和多媒体指令对 128 位宽压缩数据运算数进行运算。

[0052] 术语“寄存器”在本文中被用于指示处理器上的存储位置，这多个位置被用作用于标识运算数的宏指令的一部分。换言之，本文所指的寄存器是从处理器外部（从程序员角度）可见的寄存器。然而，实施例的寄存器在含义上不应限于特定类型的电路。相反，实施例的寄存器仅需要能存储和提供数据，并执行本文中所描述的功能。本文所描述的寄存器可利用任何数量的不同技术通过处理器中的电路来实现，这些不同技术诸如专用物理寄存器、利用寄存器重命名的动态分配物理寄存器、专用和动态分配物理寄存器的组合等。在一个实施例中，整数寄存器存储 32 位整数数据。一个实施例的寄存器文件还包含用于压缩数据的 16 个 XMM 和通用寄存器、8 个多媒体（例如“EM64T”加法）多媒体 SIMD 寄存器。对于以下讨论，寄存器应被理解为设计成保存压缩数据的数据寄存器，诸如来自美国加利福尼亚州圣克拉拉市的英特尔公司的启用了 MMX 技术的微处理器的 64 位宽 MMXtm 寄存器（在一些实例中也称为“mm”寄存器）。既有整数形式又有浮点形式的这些 MMX 寄存器可与伴随 SIMD 和 SSE 指令的压缩数据元一起运算。相似地，与 SSE2、SSE3、SSE4 或更高级技术（通称为“SSEx”）有关的 128 位宽 XMM 寄存器也可被用于保存此类压缩数据运算数。在该实施例中，在存储压缩数据和整数数据时，寄存器不需要区分这两种数据类型。在一个实施例中，其它寄存器或寄存器组合可被用于存储 256 位或更多数据。

[0053] 在以下附图的示例中，描述了多个数据运算数。图 3A 示出了根据本发明一个实施例的多媒体寄存器中的各种压缩数据类型表示。图 3A 示出 128 位宽运算数的压缩字节 310、压缩字 320 以及压缩双字 (dword)330 的数据类型。该示例的压缩字节格式 310 是 128 位长度，且包含 16 个压缩字节数据元。一个字节在此被定义为 8 位数据。每个字节数据元的信息被存储在字节 0 的位 7 到位 0、字节 1 的位 15 到位 8、字节 2 的位 23 到位 16、以及最终字节 15 的位 120 到位 127 中。因此，该寄存器中的所有可用位均被使用。该存储布置提高了处理器的存储效率。此外，在 16 个数据元被访问的情况下，现可对 16 个数据元并行地执行一个运算。

[0054] 一般而言，数据元是与同样长度的其它数据元一起存储于单个寄存器或存储器位置中的单个数据片段。在与 SSEx 技术有关的压缩数据序列中，存储于 XMM 寄存器中的数据元数量为 128 位除以单个数据元的位长度。相似地，在与 MMX 和 SSE 技术有关的压缩数据序列中，存储于 MMX 寄存器中的数据元数量为 64 位除以单个数据元的位长度。虽然图 3A 中所示的数据类型是 128 位长度，但本发明的实施例也可运算 64 位宽或其它大小的运算数。该示例的压缩字格式 320 是 128 位长度，且包含 8 个压缩字数据元。每个压缩字包含 16 个信息位。图 3A 的压缩双字格式 330 是 128 位长，且包含四个压缩双字数据元。每个压缩双字数据元包含 32 位信息。压缩四倍长字 (quadword) 是 128 位长度，且包含两个压缩的四倍长字数据元。

[0055] 图 3B 示出替代的寄存器内数据存储格式。每个压缩数据可包括一个以上独立数据元。示出了三种压缩数据格式 :半压缩 (packed half)341、单压缩 (packed single)342 以及双压缩 (packed double)343。半压缩 341、单压缩 342 以及双压缩 343 的一个实施例包含定点 (fixed-point) 数据元。对于一个或更多个半压缩 341 的替代实施例而言，单压

缩 342 和双压缩 343 可包含浮点数据元。半压缩 341 的一个替代实施例是包含 8 个 16 位数据元的 128 位长度。单压缩 342 的一个实施例是 128 位长度，且包含 4 个 32 位数据元。双压缩 343 的一个实施例是 128 位长度，且包含 2 个 64 位数据元。将理解，此类压缩数据格式可进一步被扩展为其它寄存器长度，例如 96 位、160 位、192 位、224 位、256 位或更长。

[0056] 图 3C 示出根据本发明一个实施例的多媒体寄存器中的各种有符号和无符号压缩数据类型表示。无符号压缩字节表示 344 示出了 SIMD 寄存器中的无符号压缩字节的存储。每个字节数据元的信息被按位存储于字节 0 的位 7 到位 0、字节 1 的位 15 到位 8、字节 2 的位 23 到位 16 以及最终字节 15 的位 120 到位 127 中。因此，该寄存器中的所有可用位均被使用。该存储布置可提高处理器的存储效率。此外，在 16 个数据元被访问的情况下，现可对 16 个数据元并行地执行一个运算。有符号压缩数据表示 345 示出了有符号压缩字节的存储。注意，每个字节数据元的第 8 位是符号指示符。无符号压缩字表示 346 示出了字 7 到字 0 如何被存储于 SIMD 寄存器中。有符号压缩字表示 347 与无符号压缩字寄存器内表示 346 相似。注意，每个字数据元的第 16 位是符号指示符。无符号压缩双字表示 348 示出双字数据元如何被存储。有符号压缩双字表示 349 与无符号压缩双字寄存器内表示 348 相似。注意，必需的符号位是每个双字数据元的第 32 位。

[0057] 图 3D 是与可从美国加利福尼亚州圣克拉拉市的英特尔公司的万维网 intel.com/design/litcentr 上获得的“IA-32 英特尔体系结构软件开发者手册卷 2 :指令集参考 (IA-32 Intel Architecture Software Developer’s Manual Volume 2:Instruction Set Reference)”中描述的运算码格式类型相对应的具有 32 或更多位的运算编码（运算码）格式 360 以及寄存器 / 存储器运算数寻址模式的一个实施例的描述。在一个实施例中，移位和异或运算可通过一个或更多个字段 361 和 362 来编码。对每个指令可标识多达两个运算数位置，包括多达两个源运算数标识符 364 和 365。对于移位和异或指令的一个实施例而言，目的地运算数标识符 366 与源运算数标识符 364 相同，而在其它实施例中，它们不同。对于替代实施例，目的地运算数标识符 366 与源运算数标识符 365 相同，而在其它实施例中它们不同。在移位和异或指令的一个实施例中，由源运算数标识符 364 和 365 标识的源运算数之一被移位和异或运算的结果盖写，而在其它实施例中，标识符 364 与源寄存器相对应，而标识符 365 与目的地寄存器元相对应。对于移位和异或指令的一个实施例而言，运算数标识符 364 和 365 可被用于标识 32 位或 64 位源和目的地运算数。

[0058] 图 3E 是具有 40 位或更多位的另一替代运算编码（运算码）格式 370 的描述。运算码格式 370 与运算码格式 360 相对应，且包括可任选的前置字节 378。移位和异或运算的类型可通过一个或更多个字段 378、371 以及 372 来编码。每个指令的多达两个运算数位置可通过源运算数标识符 374 和 375 且通过前置字节 378 来标识。对于移位和异或指令的一个实施例而言，前置字节 378 可被用于标识 32 位或 64 位源和目的地运算数。对于移位和异或指令的一个实施例而言，目的地运算数标识符 376 与源运算数标识符 374 相同，而在其它实施例中，它们不同。对于替代实施例，目的地运算数标识符 376 与源运算数标识符 375 相同，而在其它实施例中它们不同。在一个实施例中，对由运算数标识符 374 和 375 标识的运算数之一移位并使其与由运算数标识符 374 和 375 标识的另一运算数进行异或运算被该移位和异或运算的结果覆盖，而在其它实施例中，对由标识符 374 和 375 标识的运算数进行的移位和异或运算的结果被写入另一寄存器中的另一数据元。运算码格式 360 和 370 允许

部分地由 MOD 字段 363 和 373 以及可任选的缩放 - 指数 - 基 (scale-index-base，SIB) 和位移字节指定的寄存器到寄存器寻址、存储器到寄存器寻址、通过寄存器的寄存器寻址、直接寄存器寻址、寄存器到存储器寻址。

[0059] 接着转到图 3F，在一些替代实施例中，64 位单指令多数据 (SIMD) 算术运算可通过协处理器数据处理 (CDP) 指令来执行。运算编码（运算码）格式 380 描述具有 CDP 运算码字段 382 和 389 的一个此类 CDP 指令。对于移位和异或运算的替代实施例而言，CDP 指令的类型可通过一个或更多个字段 383、384、387 以及 388 来编码。对每个指令可标识多达三个运算数位置，包括多达两个源运算数标识符 385 和 390 以及一个目的地运算数标识符 386。协处理器的一个实施例可对 8、16、32 以及 64 位值进行运算。对于一个实施例而言，对浮点数据元执行移位和异或运算。在一些实施例中，利用选择字段 381，可有条件地执行移位和异或指令。对于一些移位和异或指令，源数据大小可通过字段 383 来编码。在移位和异或指令的一些实施例中，零 (Z)、负 (N)、进位 (C) 以及溢出 (V) 检测可在 SIMD 字段上进行。对于一些指令，饱和类型可通过字段 384 来编码。

[0060] 图 4 是根据本发明的用于对压缩数据运算数执行移位和异或运算的逻辑的一个实施例的框图。可实现本发明的实施例以作用于诸如上述的各种类型的运算数。为简单起见，以下讨论和示例是在用于处理数据元的移位和异或指令的背景下进行的。在一个实施例中，第一运算数 401 被移位器 410 移动由输入 405 指定的量。在一个实施例中，它是右移。然而在其它实施例中，移位器执行左移运算。在一些实施例中，该运算数是标量值，而在其它实施例中，它是具有多种不同数据大小和类型（例如浮点、整数）的压缩数据值。在一个实施例中，移位计数 405 是压缩（或“矢量”）值，其每个元与要由相应移位计数元件移动的压缩运算数的元相对应。在其它实施例中，该移位计数应用于第一数据运算数的所有元。此外，在一些实施例中，移位计数由该指令中的字段（诸如中值、r/m 或其它字段）指定。在其它实施例中，移位计数由该指令指示的寄存器指定。

[0061] 然后，经移位的运算数通过逻辑 420 与值 430 进行异或运算，且异或运算结果被存储于目的地存储位置（例如寄存器）425 中。在一个实施例中，异或值 430 是压缩（或“矢量”）值，其每个元与要由相应异或元件进行异或运算的压缩运算数的元相对应。在其它实施例中，异或值 430 应用于第一数据运算数的所有元。此外，在一些实施例中，异或值由该指令中的字段（诸如中值、r/m 或其它字段）指定。在其它实施例中，异或值由该指令指示的寄存器指定。

[0062] 图 5 示出根据本发明一个实施例的移位和异或指令的运算。在运算 501，如果接收到移位和异或指令，则在运算 505，将第一运算数移动移位数。在一个实施例中，它被右移。然而在其它实施例中，移位器执行左移运算。在一些实施例中，该运算数是标量值，而在其它实施例中，它是具有多种不同数据大小和类型（例如浮点、整数）的压缩数据值。在一个实施例中，移位计数值 405 是压缩（或“矢量”）值，其每个元与要由相应移位计数元件移动的压缩运算数的元相对应。在其它实施例中，该移位计数值应用于第一数据运算数的所有元。此外，在一些实施例中，移位计数值由该指令中的字段（诸如中值、r/m 或其它字段）指定。在其它实施例中，移位计数值由该指令指示的寄存器指定。

[0063] 在运算 510，经移位值与异或值进行异或运算。在一个实施例中，异或值 430 是压缩（或“矢量”）值，其每个元与要由相应异或元件进行异或运算的压缩运算数的元相对应。

在其它实施例中，异或值 430 应用于第一数据运算数的所有元。此外，在一些实施例中，该异或值由该指令中的字段（诸如中值、r/m 或其它字段）指定。在其它实施例中，该异或值由该指令指示的寄存器指定。

[0064] 在运算 515，经移位和异或运算的值被存储于一个位置。在一个实施例中，该位置是标量寄存器。在另一个实施例中，该位置是压缩数据寄存器。在另一实施例中，目的地位置也被用作源位置，诸如由指令指定的压缩数据寄存器。在其它实施例中，目的地位置是与存储初始运算数或诸如移位计数或异或值的其它值的源位置不同的位置。

[0065] 在一个实施例中，移位和异或指令可用于在各种计算机应用中执行数据去重复。数据去重复试图找出文件之间的共同数据块，以最优化磁盘存储和 / 或网络带宽。在一个实施例中，移位和异或指令通过利用诸如使用滚动散列、散列摘要（例如 SHA1 或 MD5）找出组块边界、并（利用快速 Lempel-Ziv 方案）压缩与众不同的组块的运算，可用于提高数据去重复运算的性能。

[0066] 例如，一种数据去重复算法可通过以下伪代码示出：

[0067]

```
while (p < max) {
v = (v >> 1) XOR scramble[(unsigned char)*p];
if v 具有至少 z 个尾随 0 {
ret = 1;
break;      }
   p++;
}
```

[0068] 在上述算法中，加扰表 (scramble table) 是随机 32 位常数的 256 个条目的数组，而 v 是具有数据的前 32 个字节的散列值的滚动散列。当找出组块边界时，该算法返回 ret = 1，且位置 p 表示组块边界。值 z 可以是导致良好组块检测的诸如 12-15 的常数，且可以是应用特定的。在一个实施例中，移位和异或指令可帮助上述算法以约 2 循环 / 字节的速率运算。在其它实施例中，取决于用途，移位和异或指令帮助该算法执行得更快或更慢。

[0069] 其中使用移位和异或指令的至少一个实施例可通过以下伪代码示出：

[0070]

```
while (p < max) {
v = (v << 1) XOR brefl_scramble[(unsigned char)*p];
if v 具有至少 z 个尾随 0 {
ret = 1;
break;      }
p++;
 }
```

[0071] 在上述算法中，brefl scramble 数组中的每个条目包含原始加扰数组中的相应条

目的位折算版本。在一个实施例中，上述算法左移 v 而不是右移，且 v 包含滚动散列的位折算版本。在一个实施例中，检查组块边界通过检查前导零的最小数量来执行。

[0072] 在其它实施例中，移位和异或指令可被用于其它有用的计算机运算和算法。此外，诸实施例有助于提高密集使用移位和异或运算的许多程序的性能。

[0073] 因此，公开了用于执行移位和异或指令的技术。虽然已在附图中描述和示出了某些示例性实施例，但应理解，这些实施例仅仅是为了说明而非限制宽泛的本发明，且本发明不限于所示和所描述的特定构造和布置，因为本领域技术人员在学习本公开内容时能想到各种其它变型。在诸如本技术的发展迅速且进一步进步难以预见的技术领域中，所公开的实施例的布置和细节可能如实现技术进步所促进地容易被修改，但不背离本公开的原理或所附权利要求的范围。

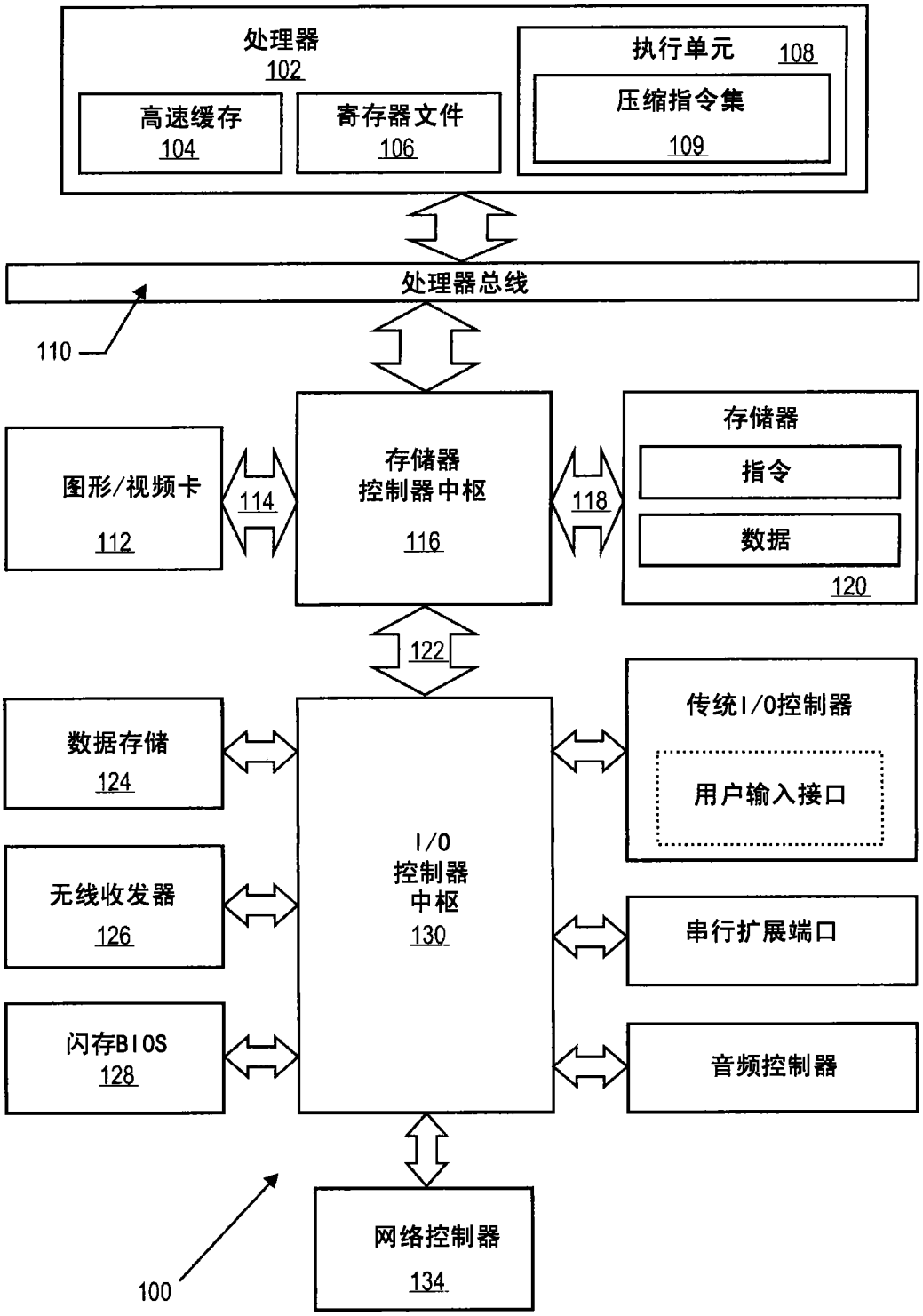

处理器
102
高速缓存
104
寄存器文件
106
执行单元
108
压缩指令集
109
处理器总线
110
图形/视频卡
112
114
存储器
控制器中枢
116
118
存储器
指令
数据
120
122
数据存储
124
无线收发器
126
闪存BIOS
128
I/O
控制器
中枢
130
传统I/O控制器
用户输入接口
串行扩展端口
音频控制器
网络控制器
134
100

图 1A

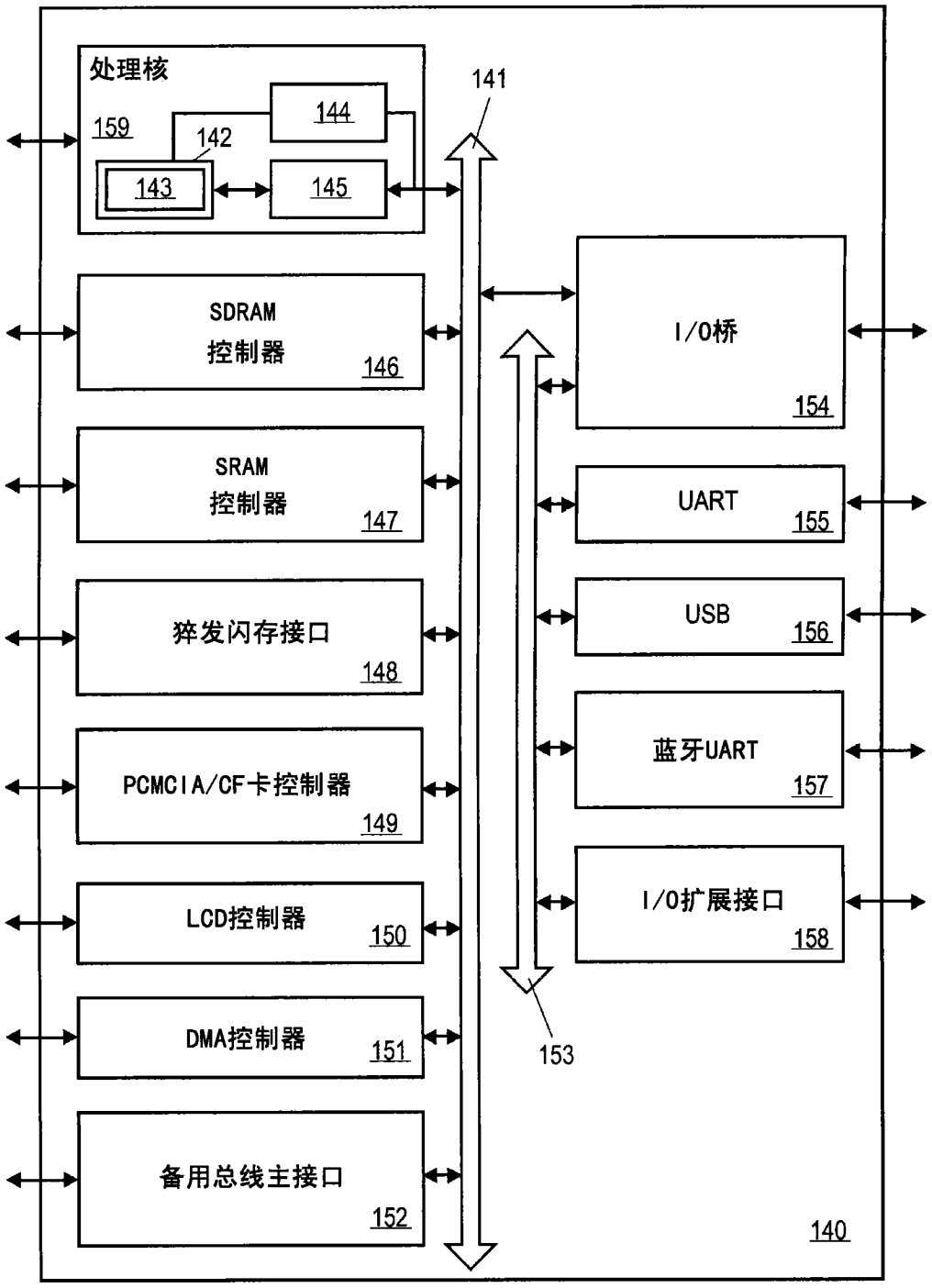

处理核
159
144
142
143
145
141
SDRAM
控制器
146
SRAM
控制器
147
猝发闪存接口
148
PCMCIA/CF卡控制器
149
LCD控制器
150
DMA控制器
151
备用总线主接口
152
I/O桥
154
UART
155
USB
156
蓝牙UART
157
I/O扩展接口
158
153
140

图 1B

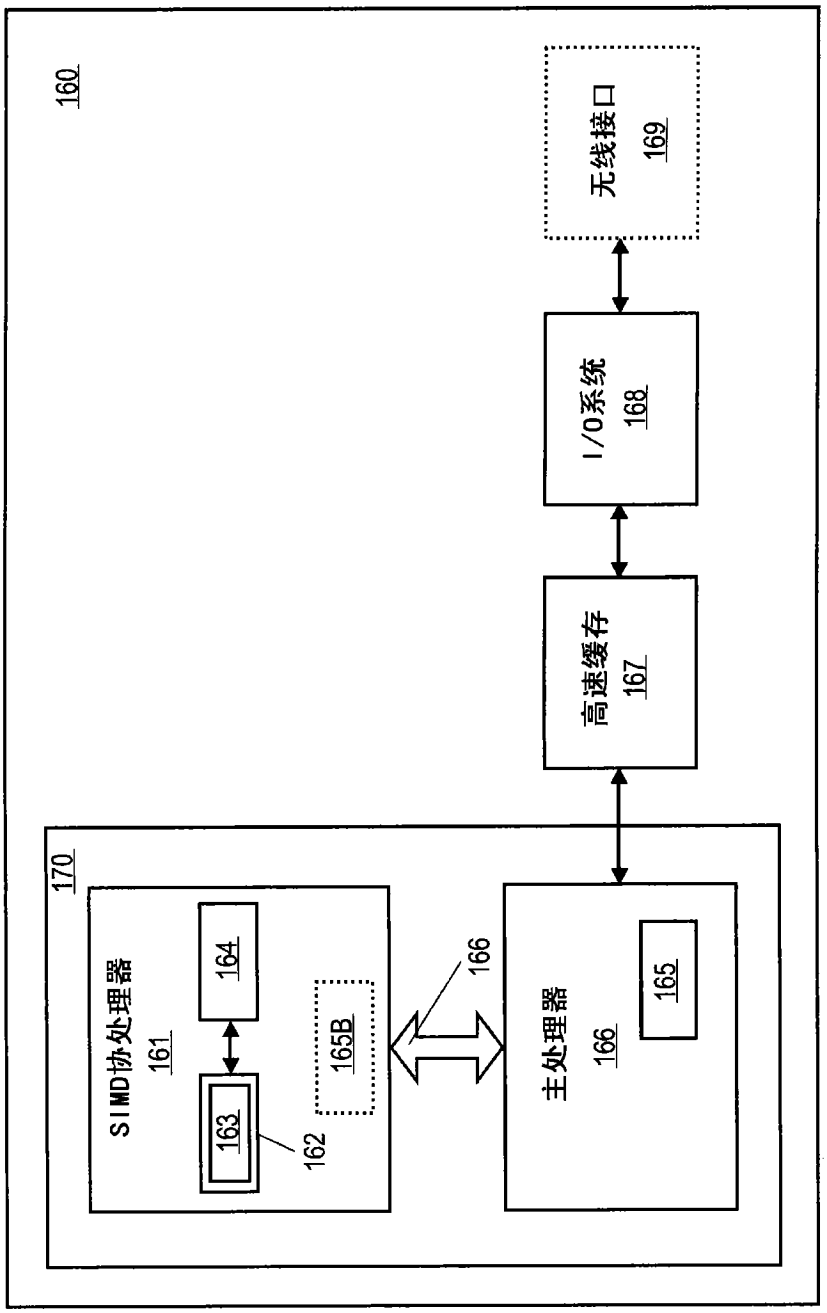

160
无线接口
169
I/O系统
168
高速缓存
167
170
SIMD协处理器
161
164
163
162
165B
166
主处理器
166
165

图 1C

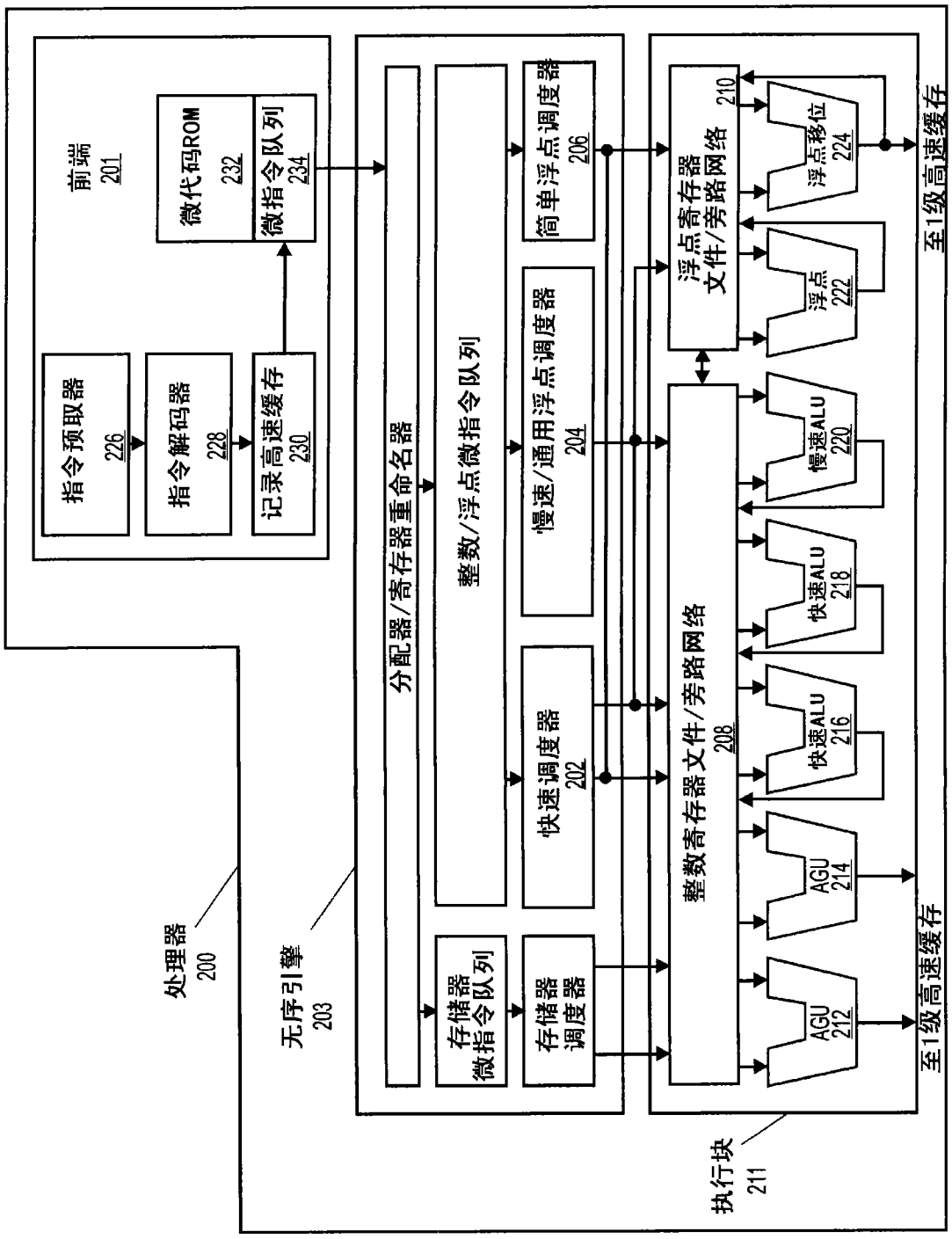

处理器
200
前端
201
指令预取器
226
指令解码器
228
记录高速缓存
230
微代码ROM
232
微指令队列
234
无序引擎
203
分配器/寄存器重命名器
整数/浮点微指令队列
存储器
微指令队列
存储器
调度器
快速调度器
202
慢速/通用浮点调度器
204
简单浮点调度器
206
执行块
211
整数寄存器文件/旁路网络
208
浮点寄存器
文件/旁路网络
210
AGU
212
AGU
214
快速ALU
216
快速ALU
218
慢速ALU
220
浮点
222
浮点移位
224
至1级高速缓存
至1级高速缓存

图 2

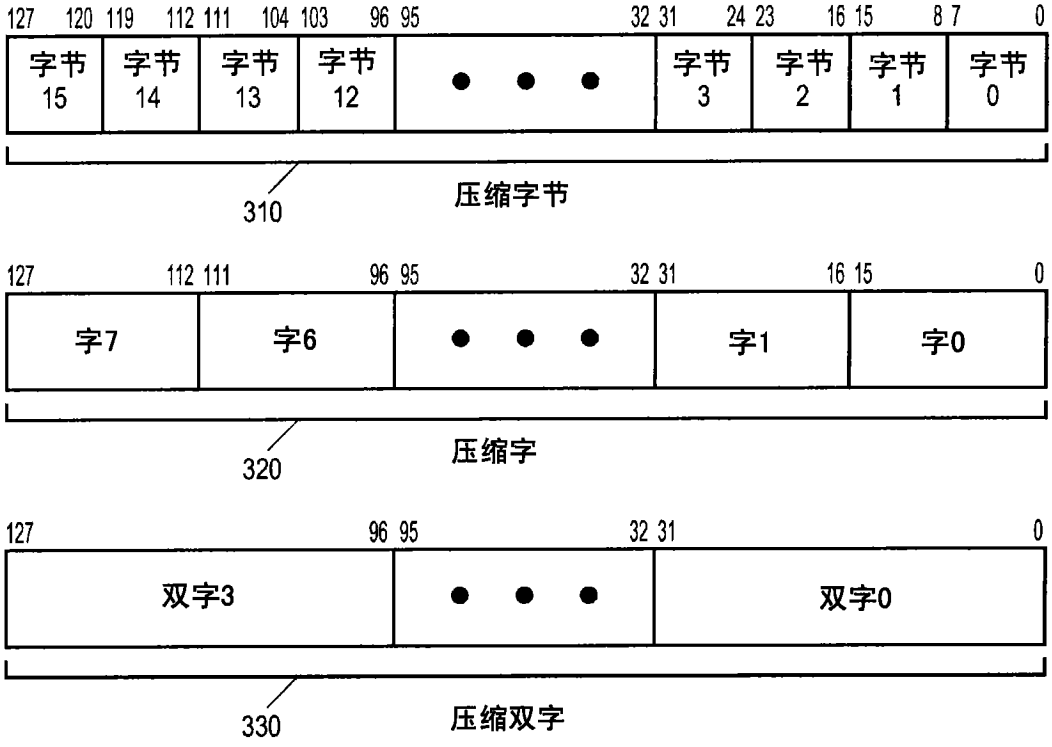

127 120 119 112 111 104 103 96 95 32 31 24 23 16 15 8 7 0
字节 15
字节 14
字节 13
字节 12
字节 3
字节 2
字节 1
字节 0
310
压缩字节
127 112 111 96 95 32 31 16 15 0
字7
字6
字1
字0
320
压缩字
127 96 95 32 31 0
双字3
双字0
330
压缩双字

图 3A

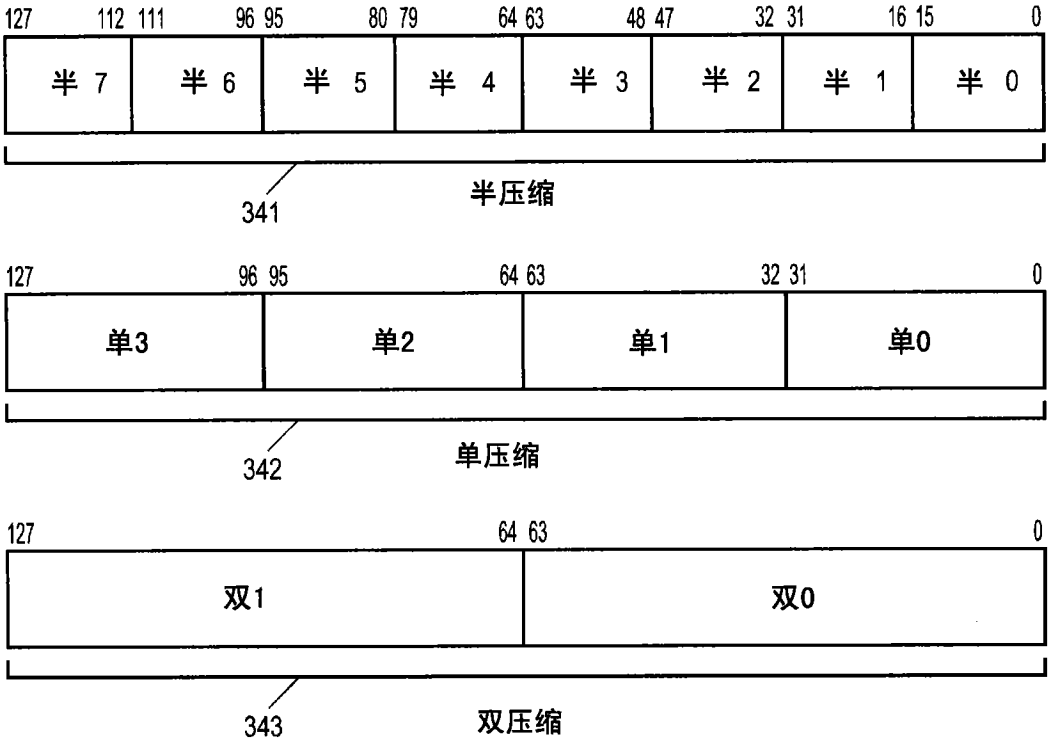
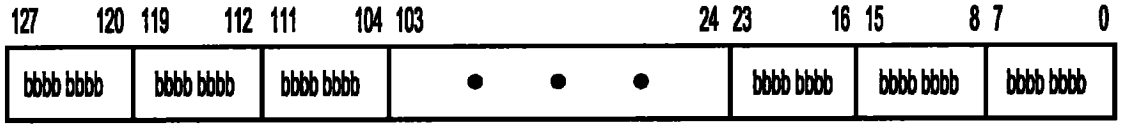
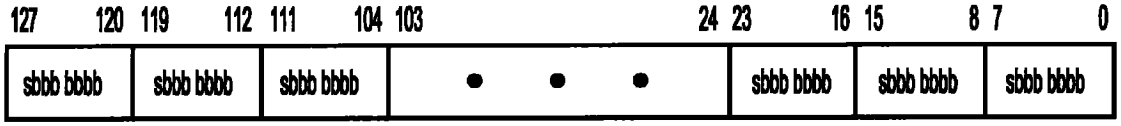

127 112 111 96 95 80 79 64 63 48 47 32 31 16 15 0
半 7
半 6
半 5
半 4
半 3
半 2
半 1
半 0
341
半压缩
127 96 95 64 63 32 31 0
单3
单2
单1
单0
342
单压缩
127 64 63 0
双1
双0
343
双压缩

图 3B

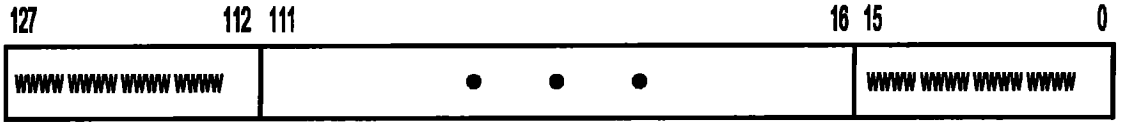


无符号压缩字节表示 344

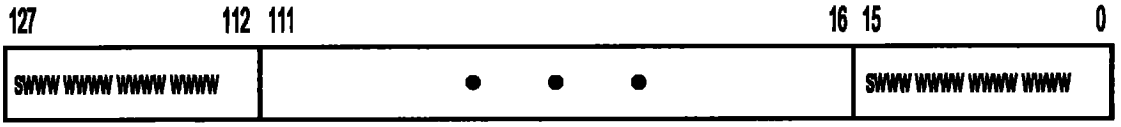
344



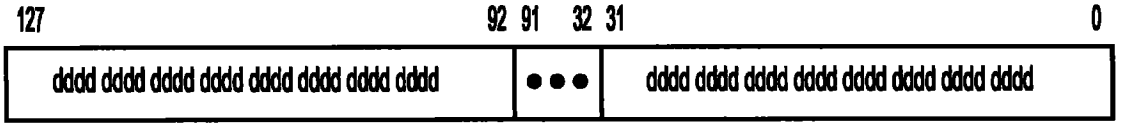
有符号压缩字节表示 345



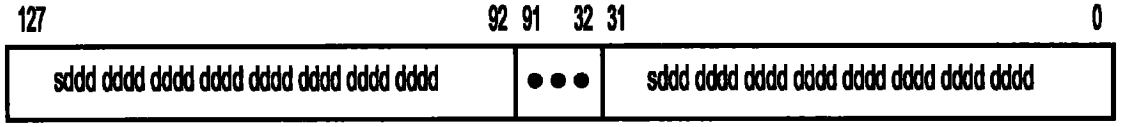
无符号压缩字表示 346



有符号压缩字表示 347



无符号压缩双字表示 348



有符号压缩双字表示 349

图 3C

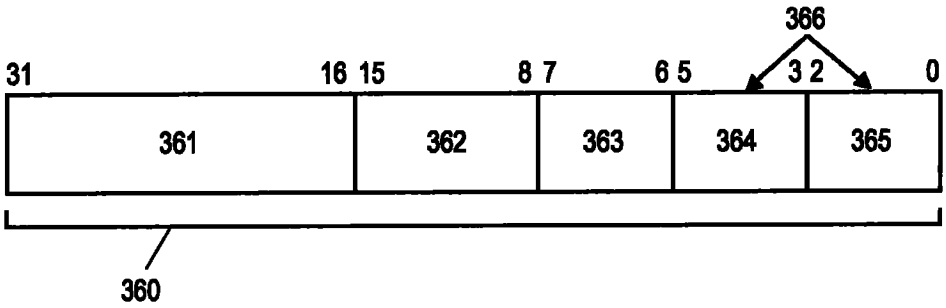

366
31
16 15
8 7
6 5
3 2
0
361
362
363
364
365
360

图 3D

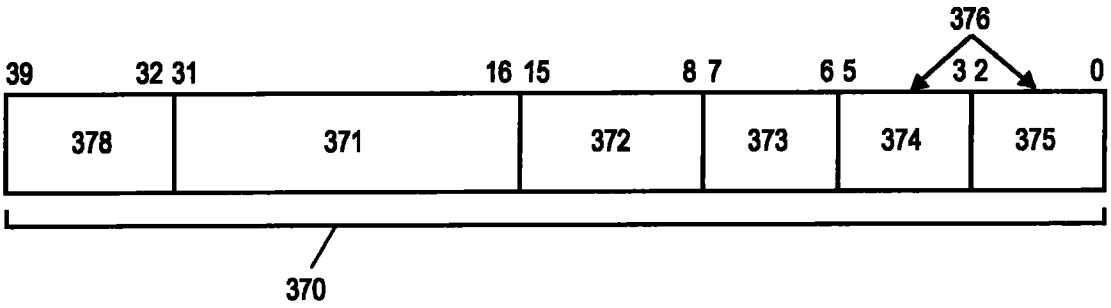

376
39
32 31
16 15
8 7
6 5
3 2
0
378
371
372
373
374
375
370

图 3E

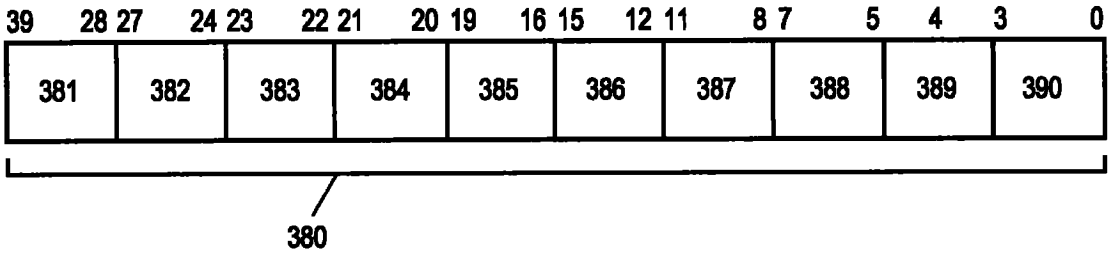

39
28 27
24 23
22 21
20 19
16 15
12 11
8 7
5
4
3
0
381
382
383
384
385
386
387
388
389
390
380

图 3F

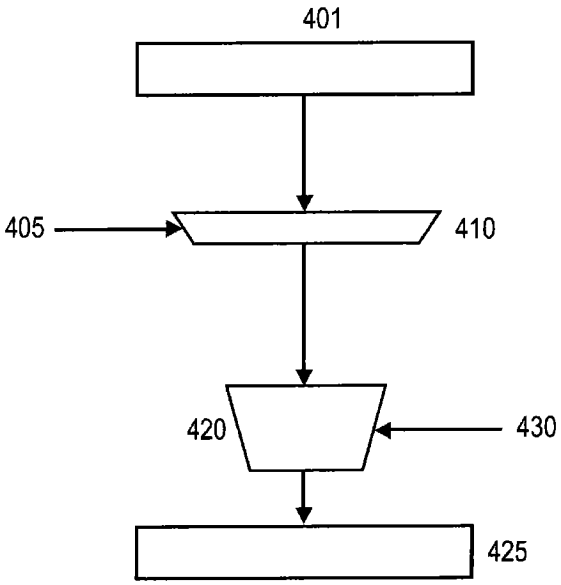

401
405
410
420
430
425

图 4

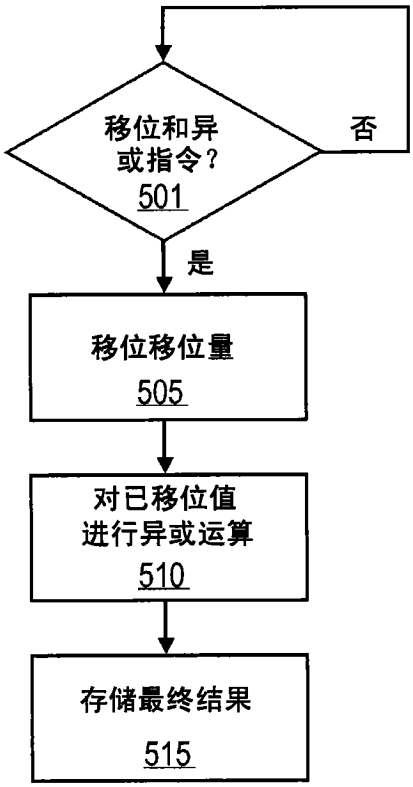

移位和异
或指令？
501
否
是
移位移位量
505
对已移位值
进行异或运算
510
存储最终结果
515

图 5