

(12) 发明专利

(10) 授权公告号 CN 1685309 B

(45) 授权公告日 2010.08.11

(21) 申请号 03822699.5

(22) 申请日 2003.09.24

(30) 优先权数据

60/413,164 2002.09.24 US

(85) PCT申请进入国家阶段日

2005.03.24

(86) PCT申请的申请数据

PCT/US2003/030089 2003.09.24

(87) PCT申请的公布数据

W02004/029793 EN 2004.04.08

(73) 专利权人 美商内数位科技公司

地址 美国特拉华州

(72) 发明人 里安·S·布杰特 夏伊·S·庭尔曼

史蒂芬·S·苏普利

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 任永武

(51) Int. Cl.

G06F 7/38 (2006.01)

G06F 7/52 (2006.01)

(56) 对比文件

US 6141637 A, 2000.10.31, 全文.

US 6317770 B1, 2001.11.13, 全文.

US 4811210 A, 1989.03.07, 全文.

US 6298366 B1, 2001.10.02, 全文.

审查员 黄捷

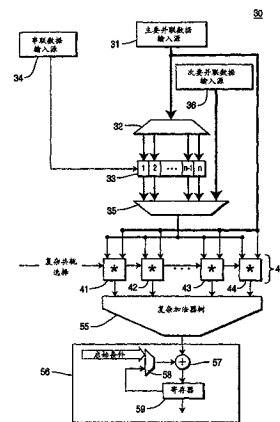
权利要求书 2 页 说明书 6 页 附图 26 页

(54) 发明名称

计算上高效数学引擎

(57) 摘要

本发明涉及在运算法使用一个单一数学引擎以使得所述数学引擎能够对所有运算法使用相同的多任务器时,用于执行许多不同形式的方法与系统。所述数学引擎包含一个选择性控制并联输出寄存器,至少一个选择性控制存储器以及多个处理组件。所述输出寄存器、存储器以及处理组件都选择性受到待使用的运算法所控制。



1. 一种计算单元,包含:

一处理器,具有多个算术逻辑单元电路,所述算术逻辑单元电路用于并行处理数据并提供处理过的数据输出,所述算术逻辑单元电路的数量定义该处理器的一计算区段的并行输入容量,且该处理器具有一加法器树,配置以增加来自所述算术逻辑单元电路的所述处理过的数据输出;

一移位寄存器,具有一并行输出以及该并行输出容量大于该处理器的该计算区段的该并行输入容量;及

用于控制该移位寄存器自该并行输出中选择性输出以匹配该处理器的该并行输入容量的装置。

2. 如权利要求 1 所述的计算单元,其特征在于进一步包含一多任务器,用于选择性接收来自该移位寄存器的输出数据,并选择性将来自该移位寄存器而作为输入的该数据提供至所述算术逻辑单元电路。

3. 如权利要求 1 所述的计算单元,其特征在于该移位寄存器同时具有串行和并行输入,并进一步包含一使能电路,配置以选择性允许该移位寄存器串行和并行输入。

4. 如权利要求 1 所述的计算单元,其特征在于进一步包含与该移位寄存器和所述算术逻辑单元电路相关联的一输入存储器,以及

一选择电路,用于根据被执行的计算的类型来选择性允许自该输入存储器至该移位寄存器及所述算术逻辑单元电路的输入。

5. 如权利要求 4 所述的计算单元,其特征在于进一步包含一第二输入存储器,其中该选择电路配置以根据被执行的计算的类型来选择性允许自该第二输入存储器至该处理器的输入。

6. 如权利要求 1、2、3、4 或 5 所述的计算单元,其特征在于该移位寄存器配置以作为:

一可选择存储器,用于接收来自至少一输入源的输入数据,且该用于控制移位寄存器的装置配置以经由多个折叠提供一可选择输出,其中各折叠具有与该处理器的该计算区段的该并行输入容量相匹配的容量,并包含位于该可选择存储器内的至少一不同位置。

7. 如权利要求 1、2、3、4 或 5 所述的计算单元,其特征在于进一步包含一累积电路,用于根据被执行的计算的类型来接收及选择性累积来自该加法器树的各输出。

8. 如权利要求 1 所述的计算单元,其特征在于进一步包含:

一存储器,接收用于复杂解的输入数据;

一储存体,储存用于复杂函数的一运算因子;

一多任务器,配置以经由该存储器或该储存体选择性接收来自该移位寄存器的输入;

所述算术逻辑单元电路配置以作为一处理阵行电路,处理来自被该存储器储存的被选择的比特位置的数据,以及自该多任务器输出的数据;及

一累积器电路,接收来自该加法器树的输出及提供一累积的复杂输出。

9. 如权利要求 8 所述的计算单元,其特征在于该储存体配置以提供一抖动因子作为执行离散富利叶转换的运算因子,其中当使用该抖动因子时,该多任务器接收来自该储存体的输出。

10. 如权利要求 1, 2, 3, 4, 5, 8 或 9 所述的计算单元,其中所述处理器配置以处理包含实部及虚部的数据。

11. 一种通信装置,包含权利要求 1,2,3,4,5,8 或 9 中的任一项所述的计算单元,配置以促进无线通信信号的处理。

12. 一种处理数据的方法,包含:

提供一处理器,其具有一加法器树以及多个算术逻辑单元电路,所述算术逻辑单元电路的数量定义该处理器的一计算区段的并行输入容量;

提供一移位寄存器,其具有 N 个并行数据输出以及该并行数据输出容量大于该处理器的该计算区段的该并行输入容量;

接收于该移位寄存器中被处理的数据;

控制该移位寄存器选择性输出数据以匹配该处理器的该并行输入容量;

使用该处理器的算术逻辑单元电路并行处理来自该移位寄存器的数据输出,并提供处理过的数据输出;及

使用该加法器树增加来自所述算术逻辑单元电路的所述处理过的数据输出。

13. 如权利要求 12 所述的方法,进一步包含使用一多任务器以选择性接收来自该移位寄存器的输出数据,并选择性提供来自该移位寄存器的该数据至所述算术逻辑单元电路。

14. 如权利要求 12 所述的方法,其中该移位寄存器同时具有串行和并行输入,且其特征在于根据被执行的数据处理计算的类型来选择性允许该移位寄存器串行和并行输入。

15. 如权利要求 12 所述的方法,其中一输入存储器与该移位寄存器和所述处理器的算术逻辑单元电路相关联,且其特征在于根据被执行的计算的类型来选择性允许自该输入存储器至该移位寄存器及所述处理器的算术逻辑单元电路的输入。

16. 如权利要求 15 所述的方法,其特征在于进一步根据被执行的计算的类型来选择性允许自一第二输入存储器至该处理器的输入。

17. 如权利要求 12,13,14,15 或 16 所述的方法,其中所述被处理的数据为包含实部及虚部的数据。

18. 如权利要求 12,13,14,15 或 16 所述的方法,该方法在一通信装置中被执行以促进无线通信信号的处理。

计算上高效数学引擎

技术领域

[0001] 本发明有关使用数学引擎来计算复杂乘数阵行的输出。更特别是,本发明为可存取执行多个数学计算的计算上高效数学引擎。

背景技术

[0002] 最新无线通信系统通常需大量数学计算来执行信号处理。该计算通常借由处理器及特定应用集成电路 (ASICs) 来执行。

[0003] 用于接收器的标准特定应用集成电路设计需执行及计算许多算法,其于规定期间通常需要许多平行相乘来完成该计算。这些算法通常包含许多矩阵对矩阵及矩阵对向量乘法,及许多离散富利叶转换 (DFT) 及快速富利叶转换 (FFT) 计算。因为乘法器占用特定应用集成电路大量空间,所以预期设计可施加相同乘法器跨越若干算法的解决方案。

[0004] 特定共享计算可被用来支持各种目前无线技术,如 WCDMA, WTT, CDMA2000, 802. 1X, TDSCDMA, FDD, TDD, 及不被呈现的其它未来系统架构。普遍被执行之一该类计算为点乘积乘法。执行点乘积计算为两矩阵间操作的标准函数。例如,执行频道估计及数据估计需点乘积计算。宽频分时双工系统中,该计算可包含主要因子快速富利叶转换计算,一矩阵乘上另一矩阵的乘法,一矩阵乘上其复杂共轭转置的乘法,及一矩阵乘上一向量的乘法。

[0005] 通常,若干点乘积计算必须借由单通信装置来执行,因此通信装置必须具有充足处理功率来支持该所需计算。目前,各算法是使用专用硬件来执行其自我数学函数。发展再使用硬件来最大化操作效率的系统是有利的。操作效率包含但不限于处理时间,执行处理的硅面积,及处理其间硅所需的功率。

[0006] 发明内容

[0007] 依据本发明,数学引擎是提供用来执行多类数学计算使硬件得以被有效使用。本发明包含一存储器,具有一并行输出,被用来储存一个或更多被选择输出于逻辑近似值的并行输出中的值。如向量的被储存值长度超过计算区段容量的例中,存储器是寻址来提供促进完成全向量的数学执行的逻辑序行中被称为折叠 (fold) 的向量部件。

[0008] 不同演算结果是借由选择性使用允许数据转移及正确数学计算来控制数学引擎操作的赋能信号而产生。当设计信号处理器时,此具有增加数学引擎执行不同类型计算的弹性,且提供节省处理器电路降低半导体固定资产所需量的优点。

[0009] 附图说明

[0010] 图 1 为本发明数学引擎的大致方块图。

[0011] 图 2A-C 显示具有其复杂共轭转置 (A) 用来计算自我相关矩阵 ($A^H A$) 的系统响应矩阵 (AH)。

[0012] 图 3A-G 显示执行 $A^H A$ 函数所需计算的图 1 的数学引擎。

[0013] 图 4A-D 显示 A^H 矩阵与该被接收信号向量 (r) 的乘法。

[0014] 图 5A-I 显示执行 $A^H r$ 函数所需计算的图 1 的数学引擎。

[0015] 图 6A-D 显示执行离散富利叶转换所需计算的图 1 的数学引擎。

[0016] 图 7A-C 显示选择性使用输入源。

[0017] 具体实施方式

[0018] 本发明可参考附图来说明,其中遍及全文的相同标号是标示相似组件。

[0019] 本发明为处理多个独立及不同算法的单数学引擎。该数学引擎可针对所有算法使用相同硬件。因为乘法器需要特定应用集成电路上显著空间,所以本发明降低特定应用集成电路所需的空间量。本发明的数学引擎亦借由较多时间使用硬件来非常有效执行所需计算。数学引擎的效率是视输入矩阵大小及处理组件数量而定。

[0020] 通常,数学引擎具有至少两输入及一输出。该输入包含一串联输入及一并联输入,其中该并联输入是与该处理组件数量一样宽。处理组件数量可被最佳化为整个向量、一部分向量、或矩阵的向量。并联及串联输入均可针对不同操作类型被加载移位寄存器或另外类型串联存取寄存器。并行输出移位寄存器为一种具有一并行输出且可快速输出被储存数据的存储器。移位寄存器的并行输出是多路传送使得可执行串联存取寄存器功能的其宽度借由处理组件数量决定的逻辑近似值,可执行具有可选择输出的并行输出移位寄存器的功能或可提供对次要并联输入的存取。主要并联输入及移位寄存器的被多路传送并行输出及次要并联输入可当作对复杂乘法器及加法器树的输入,其增加数学引擎所执行的计算效率。此使数据可针对各被执行操作尽快地被移入寄存器,亦使数据针对被有效执行的操作内步骤而再组织。

[0021] 在较佳实施例中,并行输出移位寄存器是从逻辑近似数据值输出数据,使该输出得以被用来储存被选择性并行输出至计算区段的值。被储存向量长度超过计算区段容量的例子中,并行输出移位寄存器被寻址来提供促进完成全向量的数学执行的序行中的向量部件。结果,几乎每隔时脉均产生一计算,而非多步骤来准备用于操作的数据。当被与给定长度的输出电路耦合时,并行输出移位寄存器为并行输出非折叠移位寄存器,意指其存储器储存为计算区段的数据宽度(也就是处理组件的数量)的 n 倍。

[0022] 加法器树是馈送一累加器,其可促成许多不同矩阵对矩阵及矩阵对向量乘法,及促成如 A^hA 及 A^hr 的有效计算。

[0023] 依据本发明,数学引擎所执行的点乘积计算包含但不限于频道估计及数据估计所需的多个不同类型乘法,如主要因子快速富利叶转换;一矩阵乘上另一矩阵的乘法,一矩阵乘上其复杂共轭转置的乘法,及一矩阵乘上一向量的乘法。

[0024] 参考图 1,依据本发明的数学引擎 30 简单方块图显示。数学引擎 30 包含一主要并联数据输入源 (PPDIS) 31,一多任务解讯器 32,一 n 折叠移位寄存器 33,一串联数据输入源 (SDIS) 34,一多任务器 35 及一次要并联数据输入源 (SPDIS) 36。亦被包含者为一复杂乘法器处理组件 (PE) 阵行 40,包含多个阵行组件(图标 41-44),一复杂加法器树 55 及一复杂累加器 56。复杂累加器 56 包含一起始条件输入多任务器 58,一加法器 57 及一加法器输出寄存器 59。加法器输出寄存器 59 提供一累加输出。复杂乘法器处理组件阵行 40 提供包含点乘积输出的并联复杂乘法功能。操作时,主要并联数据输入源 31 的输出包含 Q 位置。串联数据输入源 34 提供复杂接收芯片序行给移位寄存器 33。来自串联数据输入源 34 的输出是用来并联负载该移位寄存器 33。移位寄存器 33 具有 n 折叠,其中各折叠具有 Q 位置,各位置包含一复杂值。此外,其可负载具有来自主要并联数据输入源 31 的 Q 位置的值的任何折叠,或每个时脉移位零点的 X 位置。多任务器 35 接收来自移位寄存器 33 的输出及来自

次要并联数据输入源 36 的复杂 Q 位置值。

[0025] 来自多任务器 35 的输出是提供至复杂乘法器阵行 40, 其被用来提供对应输入至复杂加法器树 55。复杂加法器树 55 为一 Q 输入复杂加法器树。

[0026] 复杂加法器树 55 提供其输出至累加器 56。累加器 56 是于加总器 57 处被提供一起始条件, 其经由多任务器 58 来提供。加总器 57 的输出结果储存于输出寄存器 59 中。

[0027] 图 1 所示本发明数学引擎 30 可执行若干不同类型计算, 借此减少独立处理器的需要, 因此大为降低该数学计算所需的特定应用集成电路实际资产数量。宽频分时双工通信系统中的较佳实施例三个不同应用说明如下但不限于: 1) 自我相关矩阵 ($A^H A$) 的产生; 2) 白匹配滤波器 ($A^H r$); 及 3) 经由 Steiner 算法执行频道估计。然而, 熟悉本技术人士应了解只要不背离本发明精神及范畴, 其它算法亦可被实行。

[0028] 执行 $A^H A$ 函数的本发明一例将参考图 2A-C 及图 3A-G 来说明。此例将说明使用依据本发明的图 1 的数学引擎来执行 $A^H A$ 函数的处理。 $A^H A$ 函数为数学引擎所计算的最复杂函数之一, 因为其是计算自部分储存矩阵且该相同部分储存矩阵是以仅具有部分计算结果的不同方式存取。该操作的顺序并不以矩阵逻辑级数为基础, 而是矩阵如何借由数学引擎既存功能来配适。

[0029] 图 2A 为 A^H , A , $A^H A$ 矩阵的简单字段矩阵表示。仅 A^H 行块被储存于存储器中, 所以整个 A^H 及 A 矩阵表示为可以被转换来使用如 A 矩阵中的数据的数据虚数部分来存取数据的被储存行块上的寻址函数。最终 $A^H A$ 矩阵是压缩处理, 仿佛其被建构自 A^H 及 A 矩阵。

[0030] 因为 A^H 子块仅被储存于一单存取存储器中 (图 1 中的主要并联数据输入源 32), 所以第一步骤复制 A^H 存储器的内容进入移位寄存器 33, 其中当其仅借由被用于计算中的数据窗察觉时, 其亦可被操纵呈现第二及第三行块。因为 A^H 矩阵为扩充因子 (SF)+ 频道响应长度 (W)-1 值宽, 所以仅有 20 处理组件, 各 A^H 文字为 20 值。数据移入 20 值块称为折叠 (此例中如 $SF+W-1 = 44$, 所以具有 3 折叠)。图 3A, B 及 C 显示呈现将第一 A^H 行加载移位寄存器的首三个时脉周期。

[0031] 图 3A 中, A^H 块的第一行的第一折叠是加载移位寄存器中的第一折叠储存。图 3B 中, A^H 块的第一行的第二折叠是加载移位寄存器中的第二折叠储存。图 3C 中, A^H 块的第一行的第三折叠是加载移位寄存器中的第三折叠储存。

[0032] 将被计算的矩阵部件强调于图 2B。整行是强调, 但仅需线内的矩阵部件被用于该计算中。图 2B 显示 A^H 矩阵第一行乘上 A 矩阵第一栏, 产生 $A^H A$ 矩阵的第一值。

[0033] 因为 A^H 及 A 矩阵行及栏块大于处理组件数量 (此例中为 44 的行及栏大小), 所以以一次一折叠来计算。各折叠片段是相乘及累积于累积寄存器 59 中 (被显示于图 1 中) 直到整行被达成为止, 该时点结果是储存外接数学引擎。 A^H 矩阵借由移位寄存器中的值来表示, 而 A 矩阵值以借由采用复杂乘法器阵行 40 中的其共轭来执行的虚值反函数而被呈现于 A^H 存储器中。

[0034] 图 3D 显示 A^H 矩阵第一行的第一折叠乘上 A 矩阵第一栏的第一折叠。此乘法结果储存于累积寄存器 59 中。图 3E 显示 A^H 矩阵第一行的第二折叠乘上 A 矩阵第一栏的第二折叠。此乘法累积结果是加至先前累积值且被储存于累积寄存器 59 中。图 3F 显示 A^H 矩阵第一行的第三折叠乘上 A 矩阵第一栏的第三折叠。此乘法累积结果加至先前累积值且被储存于累积寄存器 59 中。被储存于累积寄存器 59 中的值代表 $A^H A$ 矩阵的第一位置且被外部

储存以进一步处理。此乘法采用 3 时脉,一个用于各折叠,但仅第一时脉被呈现于图 3G 中,其显示 A^H 矩阵第一行的第一折叠乘上 A 矩阵第二有效栏的第一折叠。此乘法结果储存于累积寄存器 59 中。此处理进行 3 个时脉直到该累积结果被外部储存以进一步处理为止。

[0035] 图 2C 中,可看到 16 零点如何被移位至移位寄存器 33 左侧使 A^H 矩阵第二行块可经由被用于计算窗的输入窗来看到。此处理仅采用一时脉且可被与剩余计算平行处理,借此明显节省时间。针对被用于计算窗的值中的 A^H 子块的各移位版本, A^H 矩阵子块第一行的此移位版本必须被乘上 A 矩阵子块所有有效栏。

[0036] 第一有效 A^H 行的每个折叠均被乘上被包含于可计算的输入窗中的 A 矩阵子块每个有效栏之后,针对隔一有效行做全部相同处理。该失序处理执行使 AH 行不必针对 A^H 子块的移位版本被重新加载。

[0037] 作为第二例,借由图 1 的数学引擎所执行的 AHr 函数将参考图 4A-D 及图 5A-I 做说明。 $A^H r$ 函数为整个 A^H 函数乘上被接收向量 (r)。 A^H 矩阵并非全部被储存,仅较小部份称为 A^H 行块被实际储存; A^H 矩阵中所有其它值被实际重复 A^H 行块或零点。于是, A^H 可被储存为 A^H 行块的复杂寻址方案。

[0038] 例如,宽频分时双工第三代 (3G) 数据丛集类型 1 或 3 中总共具有 $61A^H$ 行块,而从集类型 2 时槽中的总共具有 $69A^H$ 行块。实际上具有两 r 向量,此功能将针对一 r 向量来说明;此处理接着针对第二数据组向量做重复。

[0039] 图 4A 为表示 $A^H r$ 矩阵乘法的图标。 A^H 行块于 A^H 矩阵各处被重复所有剩余零点。 A^H 行块行中,四行的二个因先前处理而失效。

[0040] 计算 $A^H r$ 函数的第一步骤以 r 向量值预载数学引擎的整个移位寄存器 33。(图 1 所示) 移位寄存器 33 借由串联数据输入源 34 串联加载。 A^H 矩阵是提供于主要并联数据输入源 31。

[0041] 为了开始一步一步处理计算 $A^H r$,以 r 向量计算 A^H 矩阵第一行显示于图 4B。因为仅 A^H 向量的第一 SF+W-1 值包含非 0 值,所以仅行乘积的第一部份被计算。因为仅有 20 处理组件,所以必须以多步骤来相乘(如折叠)此例中, W 为 29,所以 SF+W-1 为 44。以处理组件数量 (20) 除以 SF+W-1 且进位成整数,则总共具有 3 折叠,或乘上 3 时脉并累积以 r 向量有效部分来计算 A^H 行有效部分的全部点乘积。图 5A,B 及 C 所示的这些前三个时脉显示一步一步处理计算此矩阵行及向量的此点乘积第一值。

[0042] 图 5A 显示第一 A^H 行的第一折叠乘上 r 向量的第一折叠,而该结果储存于累积寄存器 59 中。图 5B 显示第一 A^H 行的第二折叠乘上 r 向量的第二折叠,而该结果是添加于目前位于累积寄存器 59 中的结果。图 5C 显示第一 A^H 行的第三折叠乘上 r 向量的第三折叠,而该结果是添加于目前位于累积寄存器 59 中的结果。最终累积现在被储存为第一 $A^H r$ 计算点。

[0043] 下一步以目前被计算符号来计算 A^H 行块的下一有效行。此牵涉 r 向量的相同部分乘上图 4C 所示新 A^H 行。r 向量可保持相同而 A^H 不同行是存取于 A^H 存储器中。再次,三时脉显示于显示所有三折叠的乘法累积的图 5D,E 及 F。图 5D 显示第三 A^H 行的第一折叠乘上 r 向量的第一折叠,而该结果储存于累积寄存器 59 中。图 5E 显示第三 AH 行的第二折叠乘上 r 向量的第二折叠,而该结果是添加于目前位于累积寄存器 59 中的结果。图 5F 显示第三 A^H 行的第三折叠乘上 r 向量的第三折叠,而该结果是添加于目前位于累积寄存器 59 中

的结果。最终累积现在被储存为第二 $A^H r$ 计算点。

[0044] 下一步开始以 r 向量来计算第二 A^H 行块。此牵涉将各有效 A^H 第二行子块行乘上 r 向量。因为下一行从最后被移位 16 值, 所以被乘上 A^H 子块的 r 向量部分是不同。图 4D 中, 被牵涉于计算中的部分是强调。

[0045] 因为被储存的 A^H 行块并不移动, 所以 r 向量必须被移位来重新校准被乘上新 A^H 行块中的行的 r 向量部分。此借由使用移位寄存器 33 对 16 时脉移位 16 新值, 及下降 r 向量第一值来达成。此处理的首两个时脉是显示于第五 G 及 H 值, 其余是推论。此后, 相同如第一 A^H 行块的处理是激活, 所以此重复操作的第一时脉是显示与第一 A^H 行块的第一时脉相比。

[0046] 图 5G 显示被放入移位寄存器的第一值遗失时, 被转移入移位寄存器右侧的 r 向量的一值。图 5H 显示被放入寄存器的第二值遗失时, 被转移入移位寄存器的 r 向量的第二值。图 5I 显示第一 AH 行乘上最新移位 r 向量的第一折叠, 而该结果正如第一 AHr 行块第一计算而被储存于累积寄存器 59 中。此处理针对每一个 AH 行块继续每个有效 AH 行块行的每个折叠。

[0047] 作为第三例, 借由图 1 的数学引擎所执行的 Steiner 算法将参考图 6A-D 做说明。图标显示使用数学引擎来操作 456pt Steiner 的 3pt 离散富利叶转换。图 6A-C 显示第一离散富利叶转换计算, 各图显示不同时间。如图所示, 离散富利叶转换的首三个地址已被串联加载存储器, 所以其全部可以并联方式经由主要并联数据输入源 31 被一次存取 (图 1 中)。被用于离散富利叶转换的抖动 (Twiddle) 因子经由次要并联数据输入源 36 被输入 (图 1 中)。

[0048] 各这三个时脉中, 地址 0, 152 及 304 处的第一离散富利叶转换输入是乘上对该离散富利叶转换唯一的三个抖动因子组之一。这三个计算结果是外接图 1 的数学引擎。

[0049] 图 6A 中, 抖动组 1 的三点乘上离散富利叶转换 1 的三点, 而该结果是外部储存。第二离散富利叶转换的第一点加载存储器下一行。图 6B 中, 抖动组 2 的三点乘上离散富利叶转换 1 的三点, 而该结果是外部储存。第二离散富利叶转换的第二点加载存储器下一行。图 6C 中, 抖动组 3 的三点乘上离散富利叶转换 1 的三点, 而该结果是外部储存。第二离散富利叶转换的第三点加载存储器下一行。

[0050] 图 6D 显示下一离散富利叶转换第一点是如何被以第一抖动因子组, 而其它两组将遵循以下时脉。64 点离散富利叶转换以些微不同方式来达成, 其中各离散富利叶转换组乘上抖动组是采用储存前被一起累积的四个连续时脉。

[0051] 图 6D 中, 抖动组 1 的三点是乘上离散富利叶转换 2 的三点, 而该结果是外部储存。第三离散富利叶转换的第一点加载存储器下一行。此乘法针对每一个 3pt 离散富利叶转换被继续用于所有三个抖动因子组。其它离散富利叶转换以 8 抖动组被同样达成 8pts, 以 19 抖动组达成 19pts, 而以 64 抖动组达成 64pts。

[0052] 参考图 7A-C, 其可见选择性输入本发明数学引擎允许数学引擎执行 $A^H A$, $A^H r$ 及 Steiner 函数。如图 7A 所示, 针对 $A^H A$ 函数, 仅主要并联数据输入源输入于 $A^H A$ 操作期间被用来提供系统响应矩阵 (A^H); 系统响应矩阵乘上其复杂共轭转置值。由于使用移位寄存器的并联负载能力且因为复杂乘法阵行包含共轭其输入之一的能力, 所以各点视需多少折叠而被计算于 x 时脉循环中 (其中 $1 \leq x = n$, n 为折叠最大值 #)。

[0053] 参考图 7B, 针对 $A^H R$ 函数, 主要并联数据输入源提供系统响应矩阵 (A^H)。于是, 串联输入于 $A^H A$ 操作期间被用来提供被接收向量 (r) 的数据栏。系统响应矩阵 A^H 针对时槽被乘上该被接收向量 (r) 的数据栏。由于使用移位寄存器的串联负载能力, 所以当系统响应矩阵被提供至主要并联输入时, 数据字段是加载移位寄存器。最终向量的各组件视需多少折叠而被计算于 x 时脉循环中 (其中 $1 \leq x \leq n$, n 为折叠最大值 #)。

[0054] 如用于图 7C 所示用于 Steiner 操作 (FFT), 主要并联数据输入源提供快速富利叶转换输入数据组, 而次要并联数据输入源提供快速富利叶转换抖动因子。快速富利叶转换操作期间, m 点快速富利叶转换中, 数据组的适当 m 点借由次要并联数据输入源提供至复杂乘法阵行, 而适当快速富利叶转换抖动因子借由次要并联数据输入源提供。

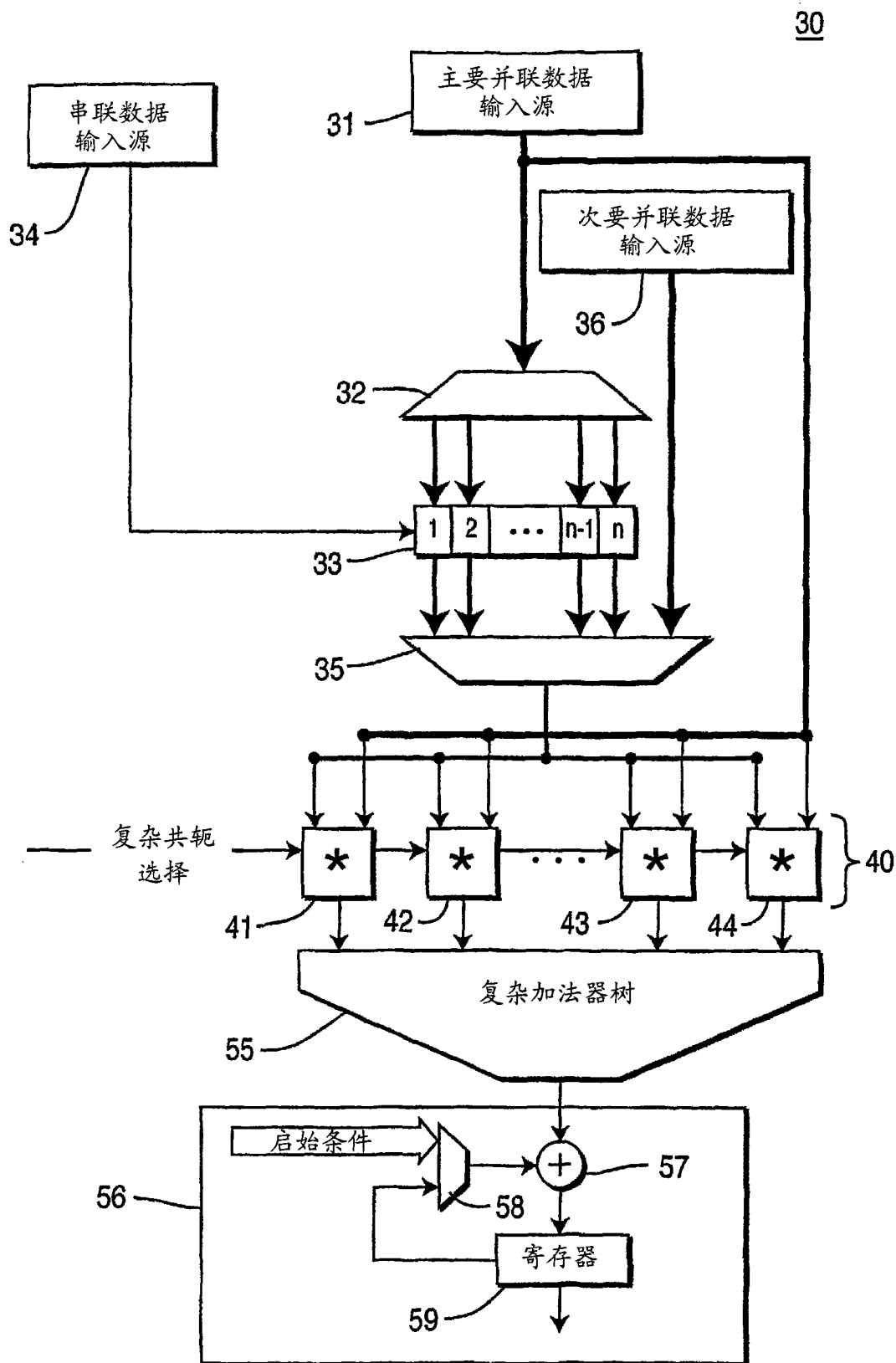


图 1

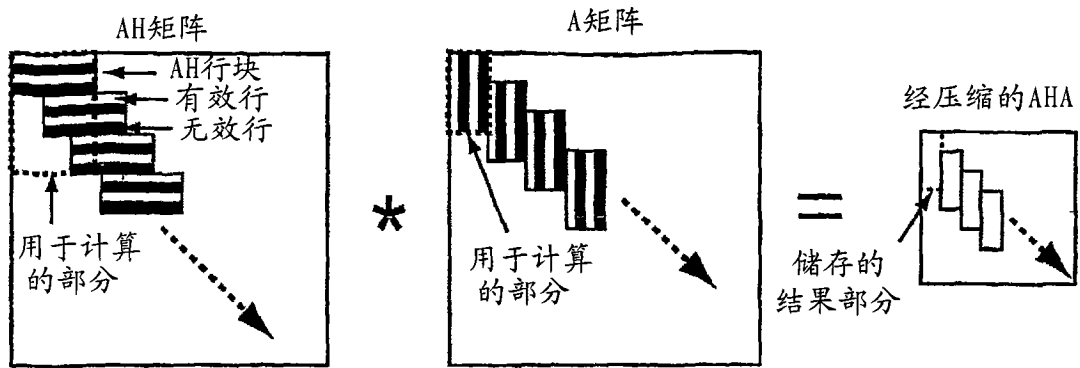


图 2A

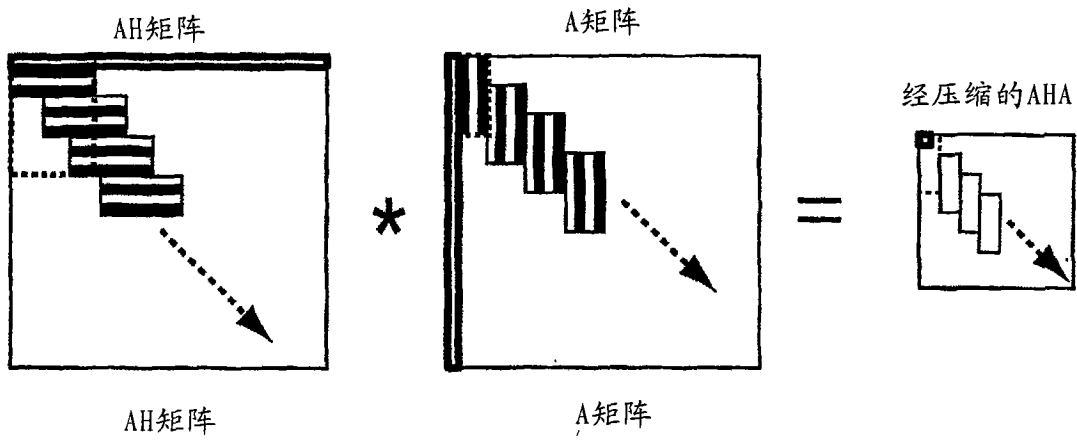


图 2B

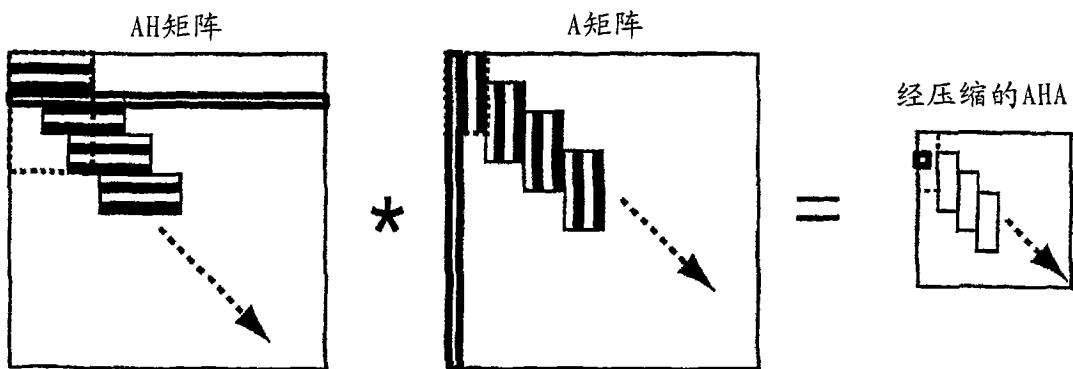
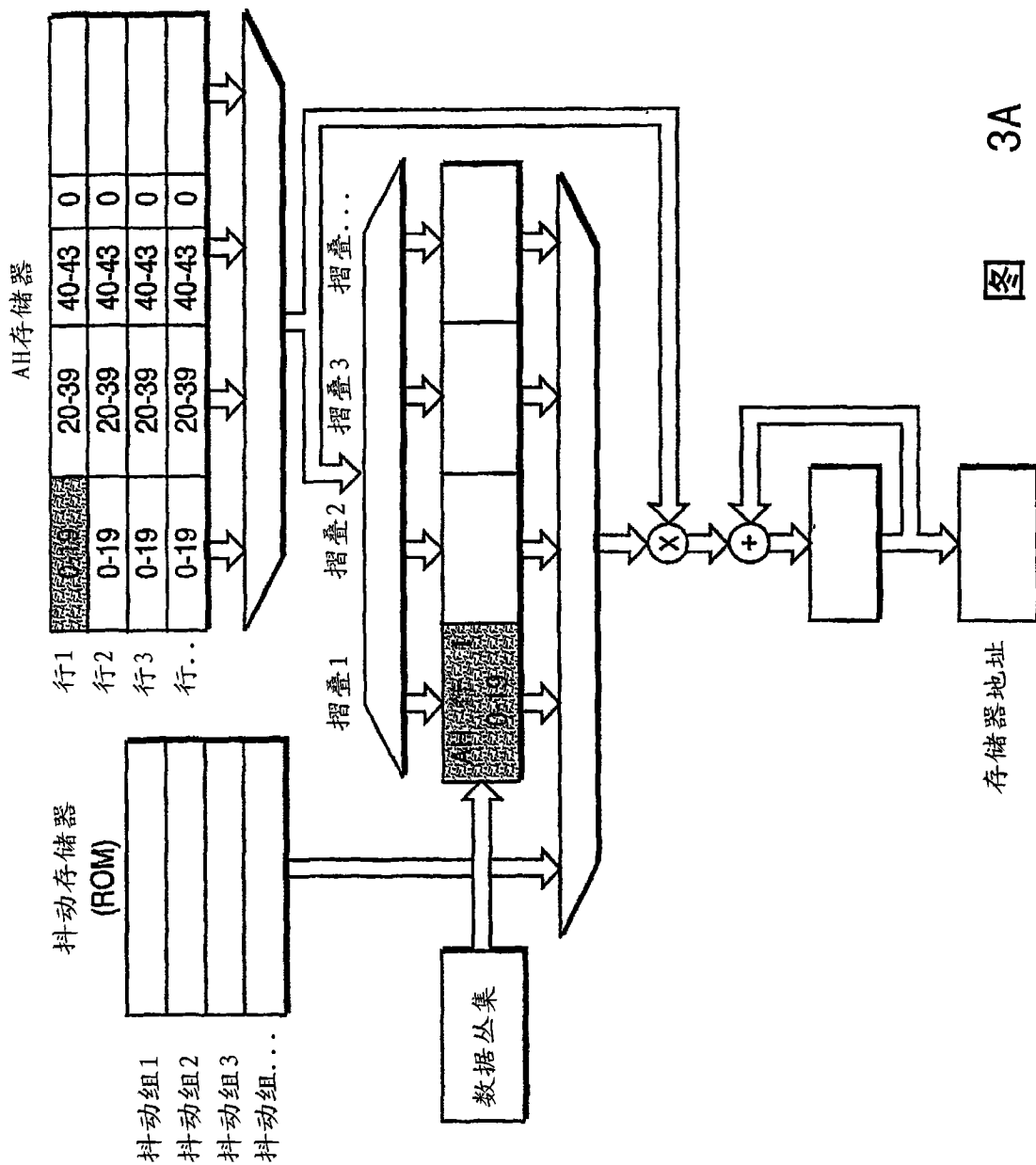
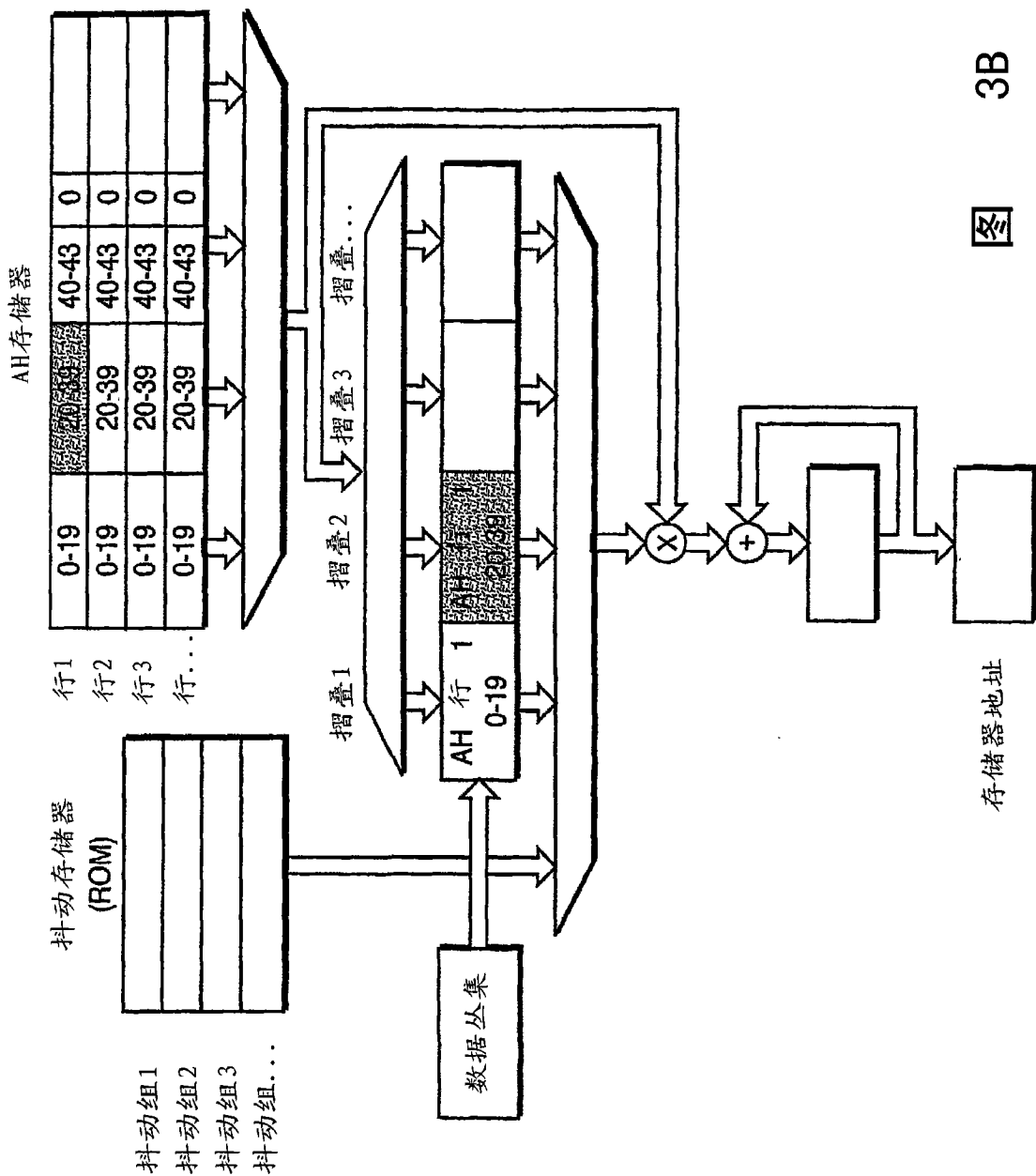


图 2C





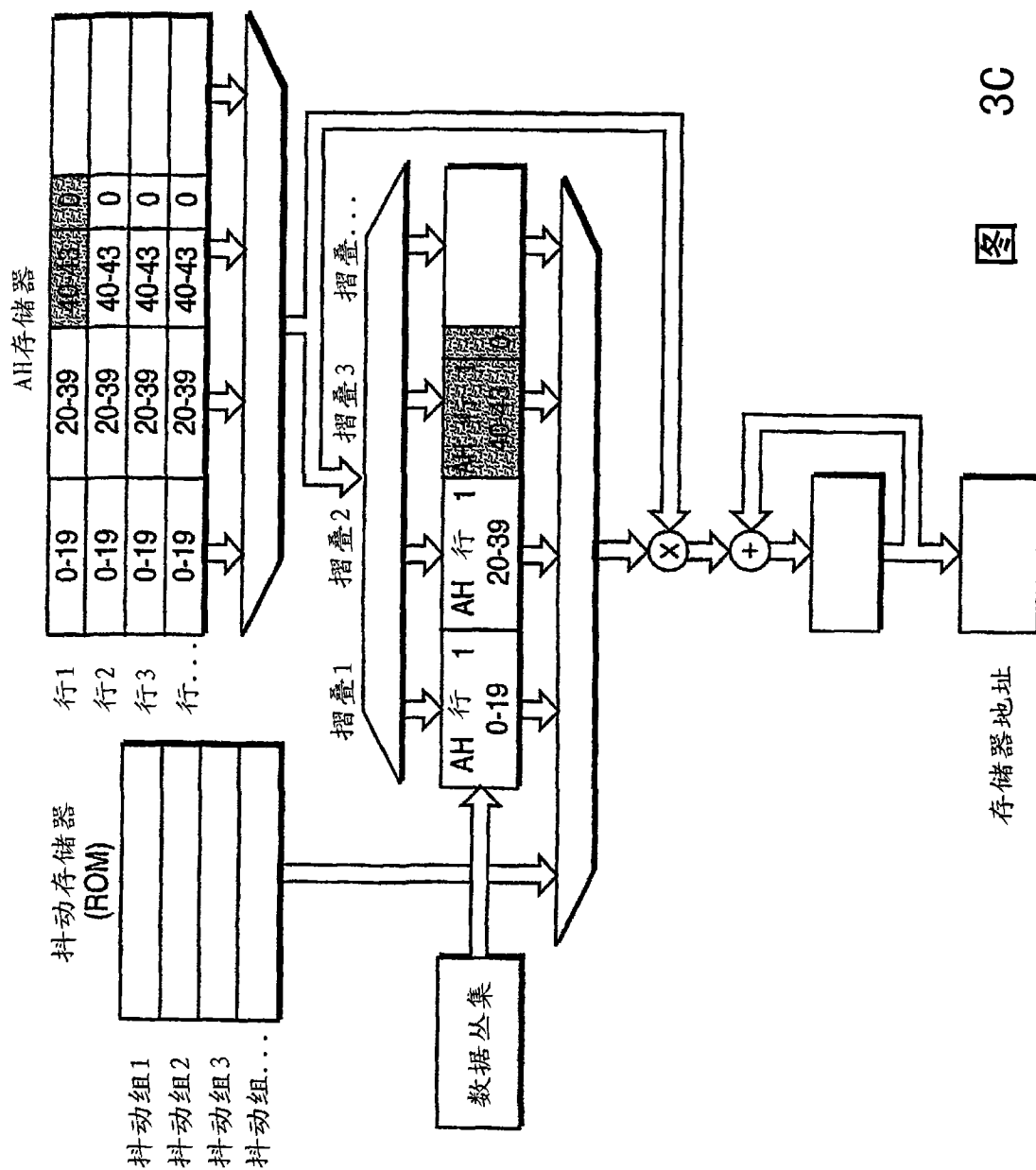
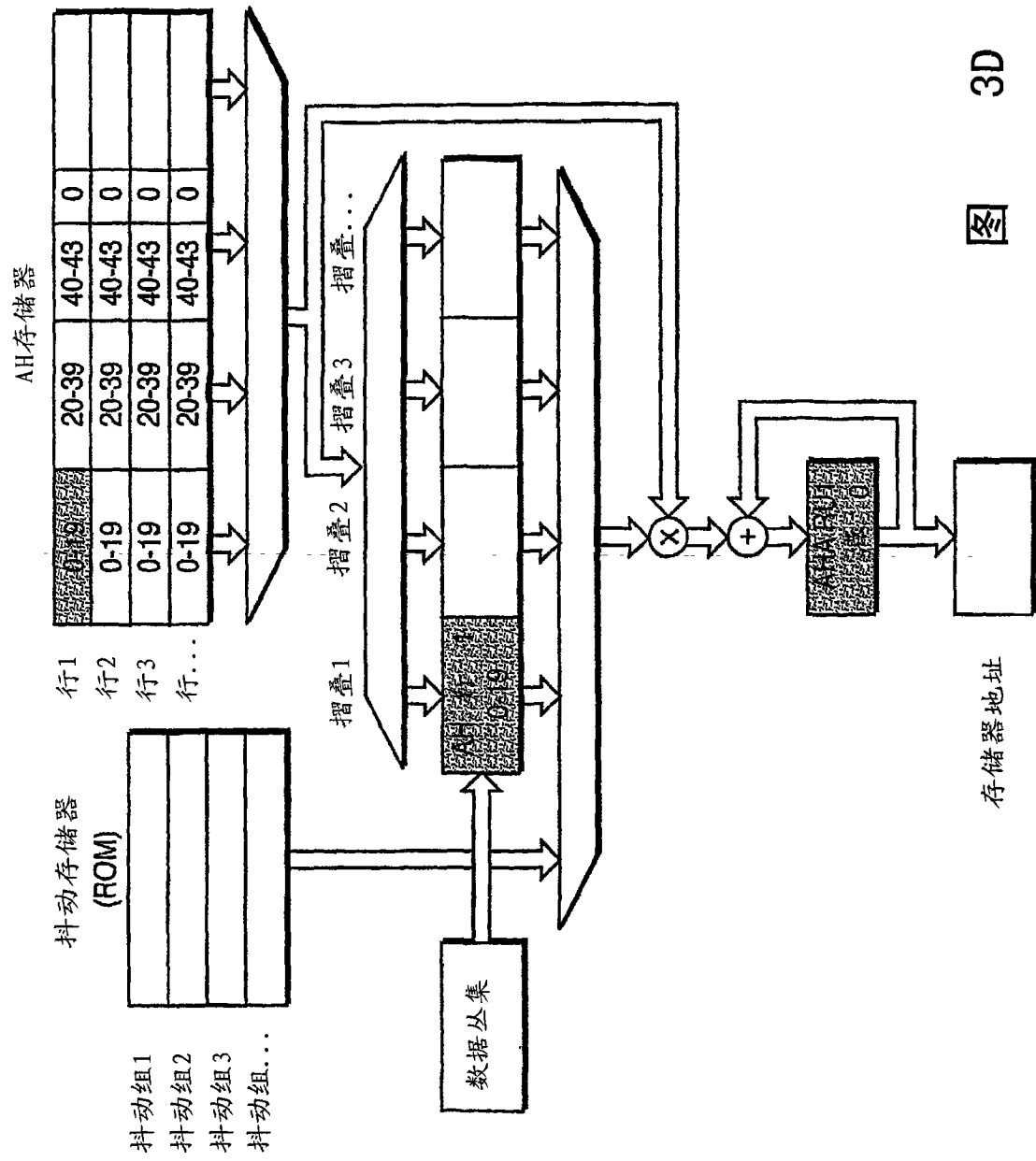


图 3C



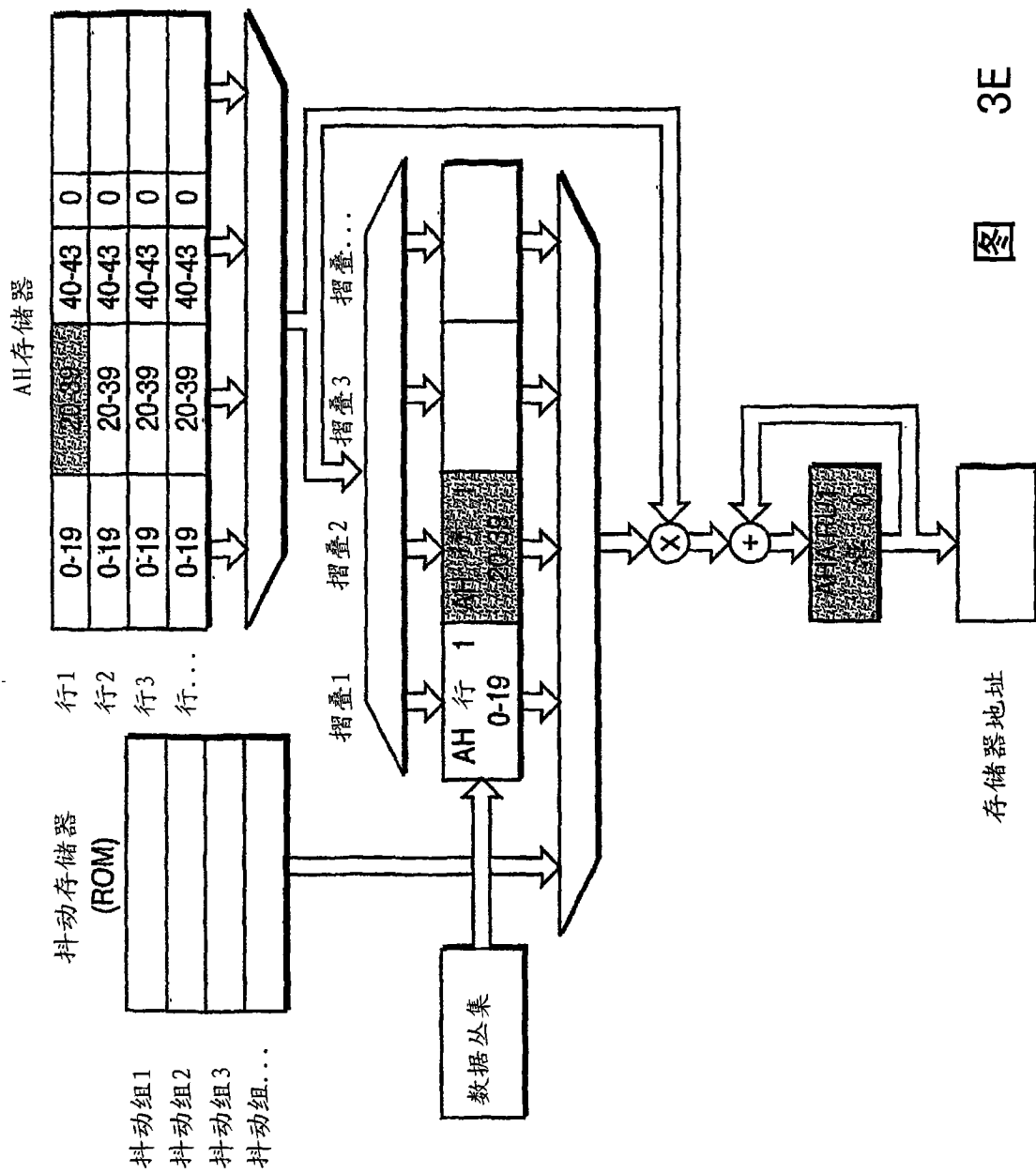


图 3E

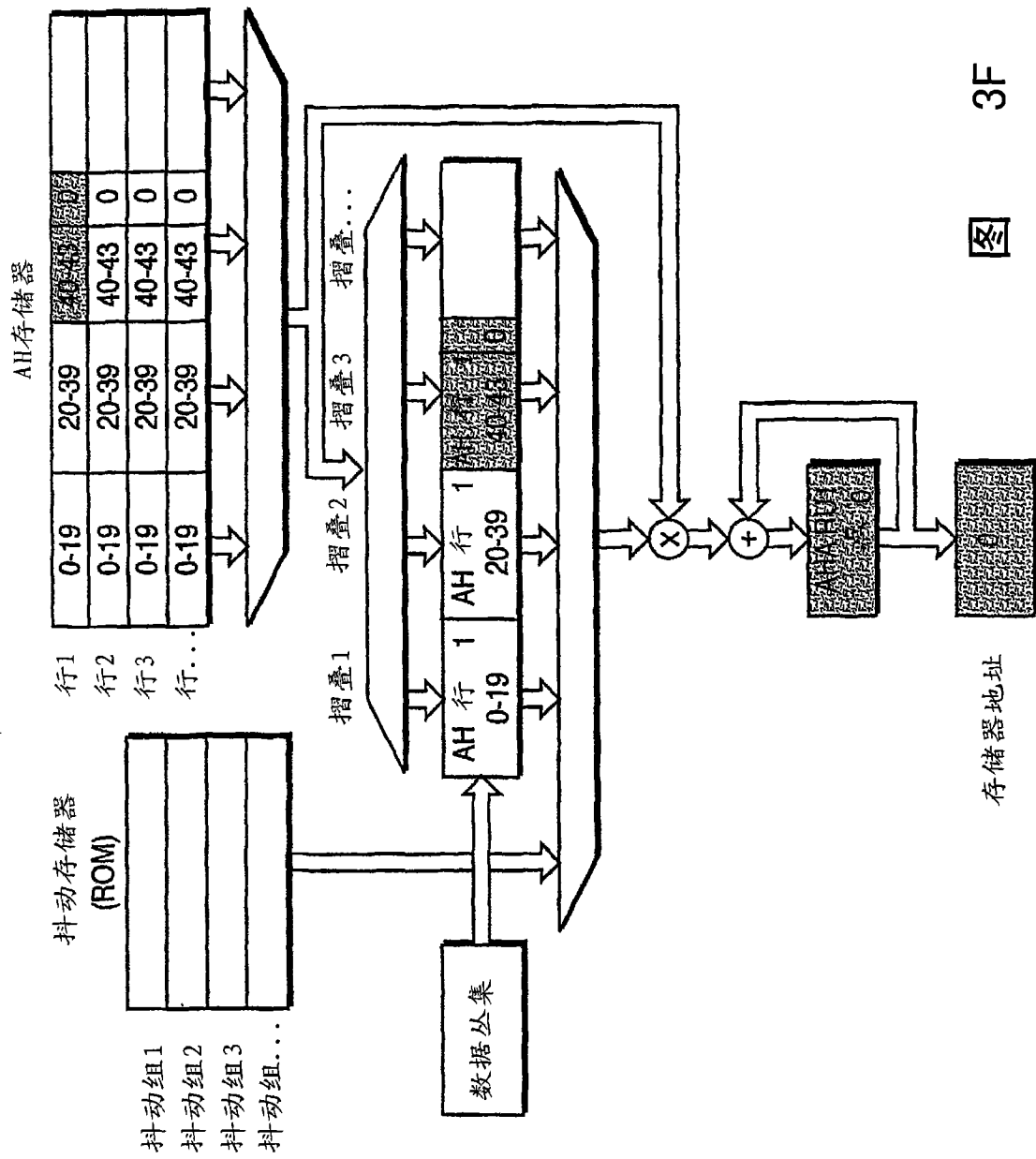
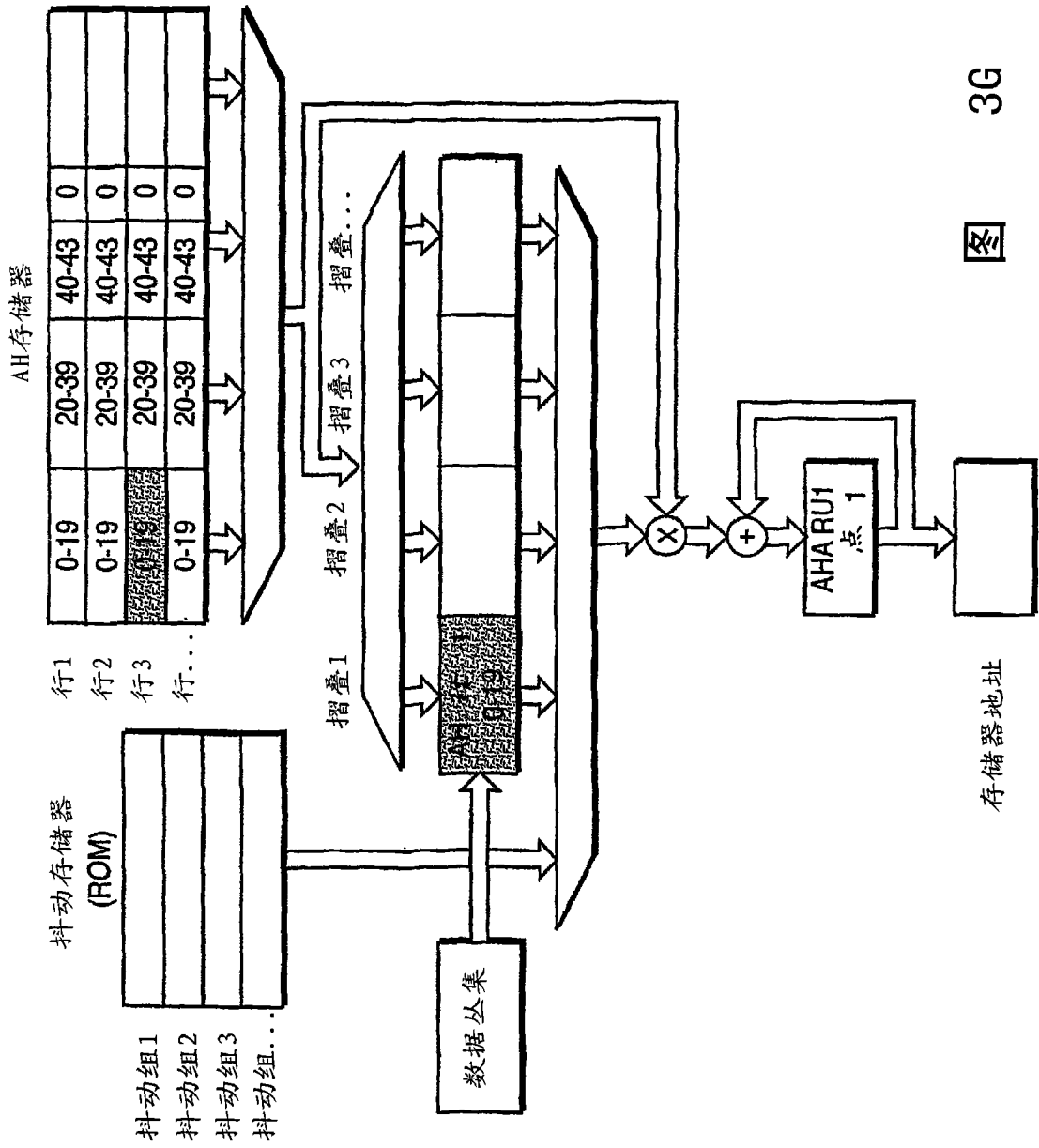


图 3F



3G
图

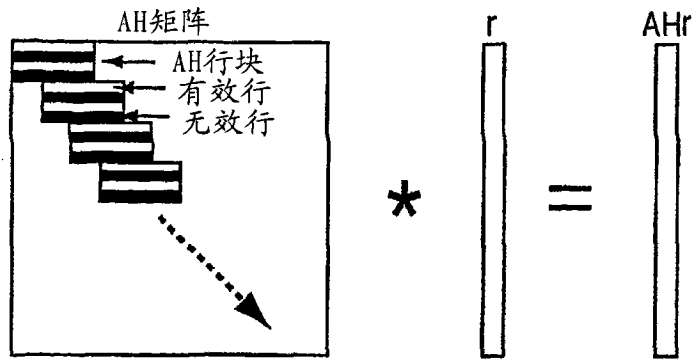


图 4A

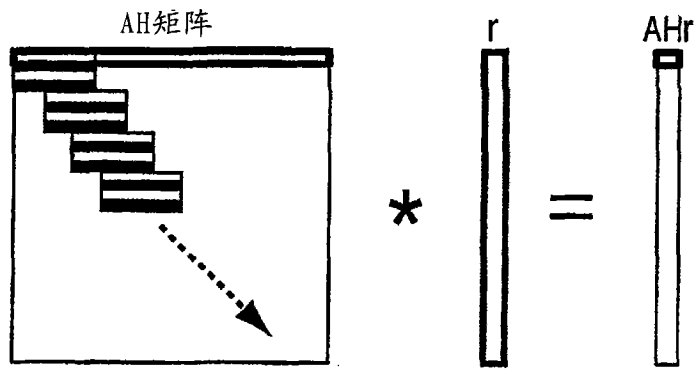


图 4B

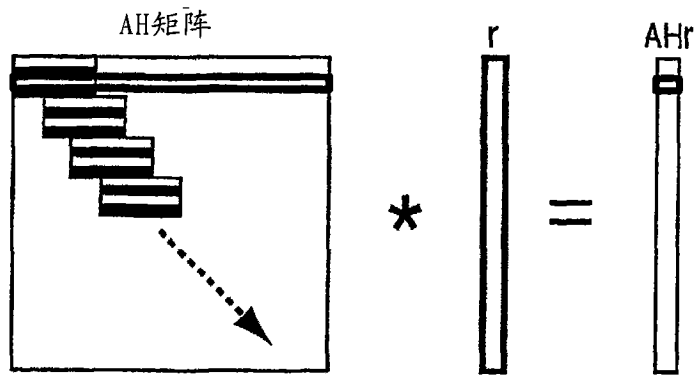


图 4C

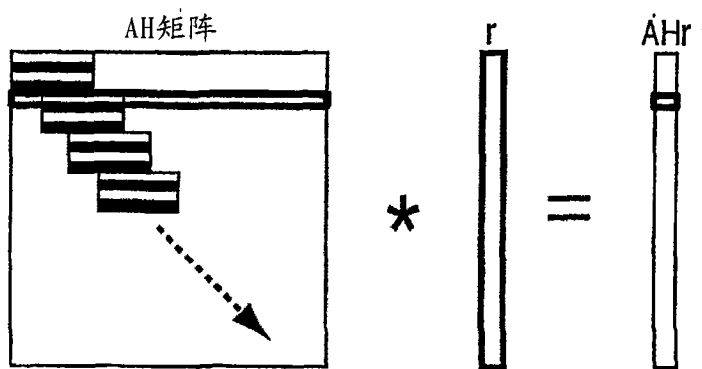


图 4D

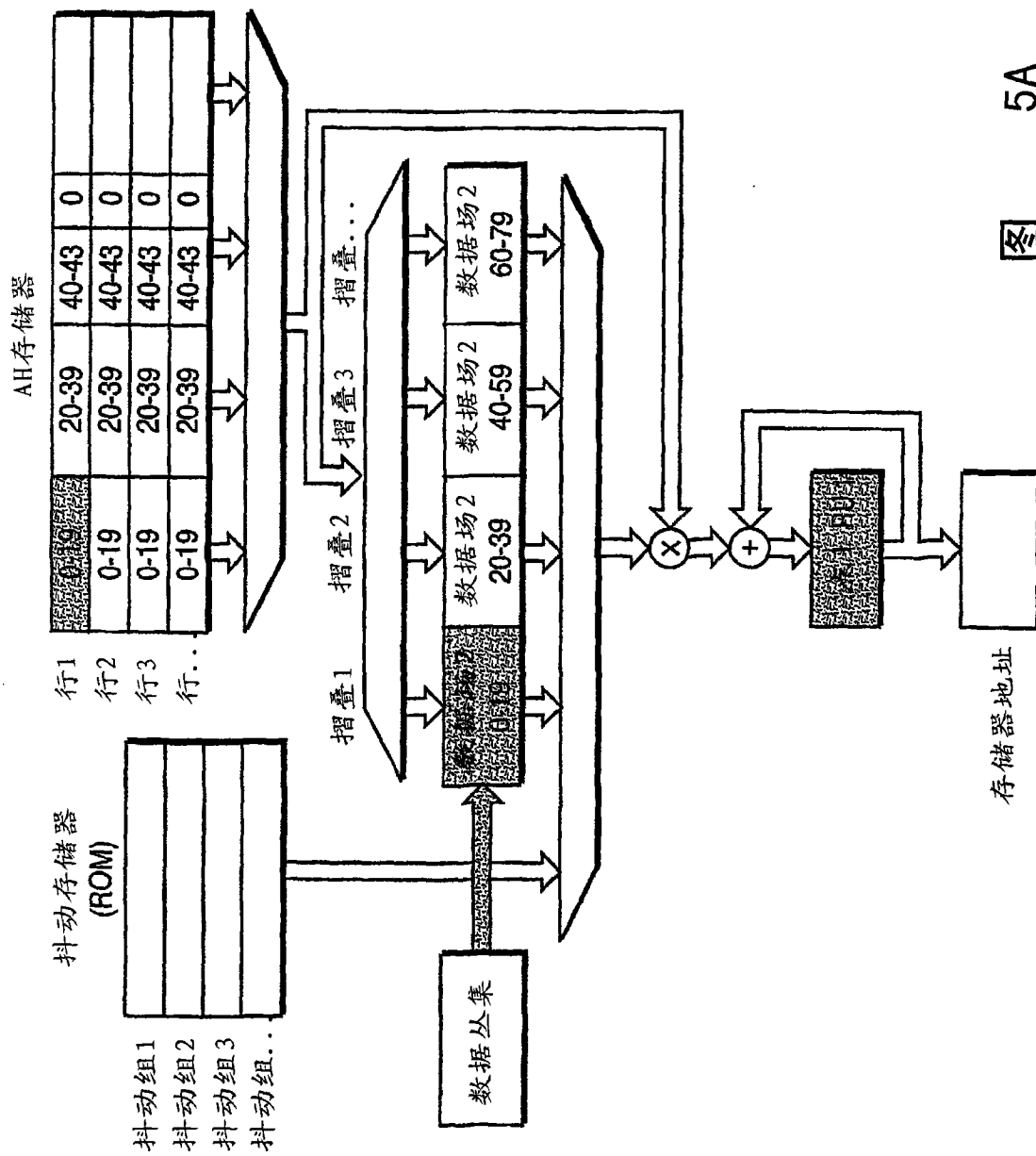


图 5A
存储器地址

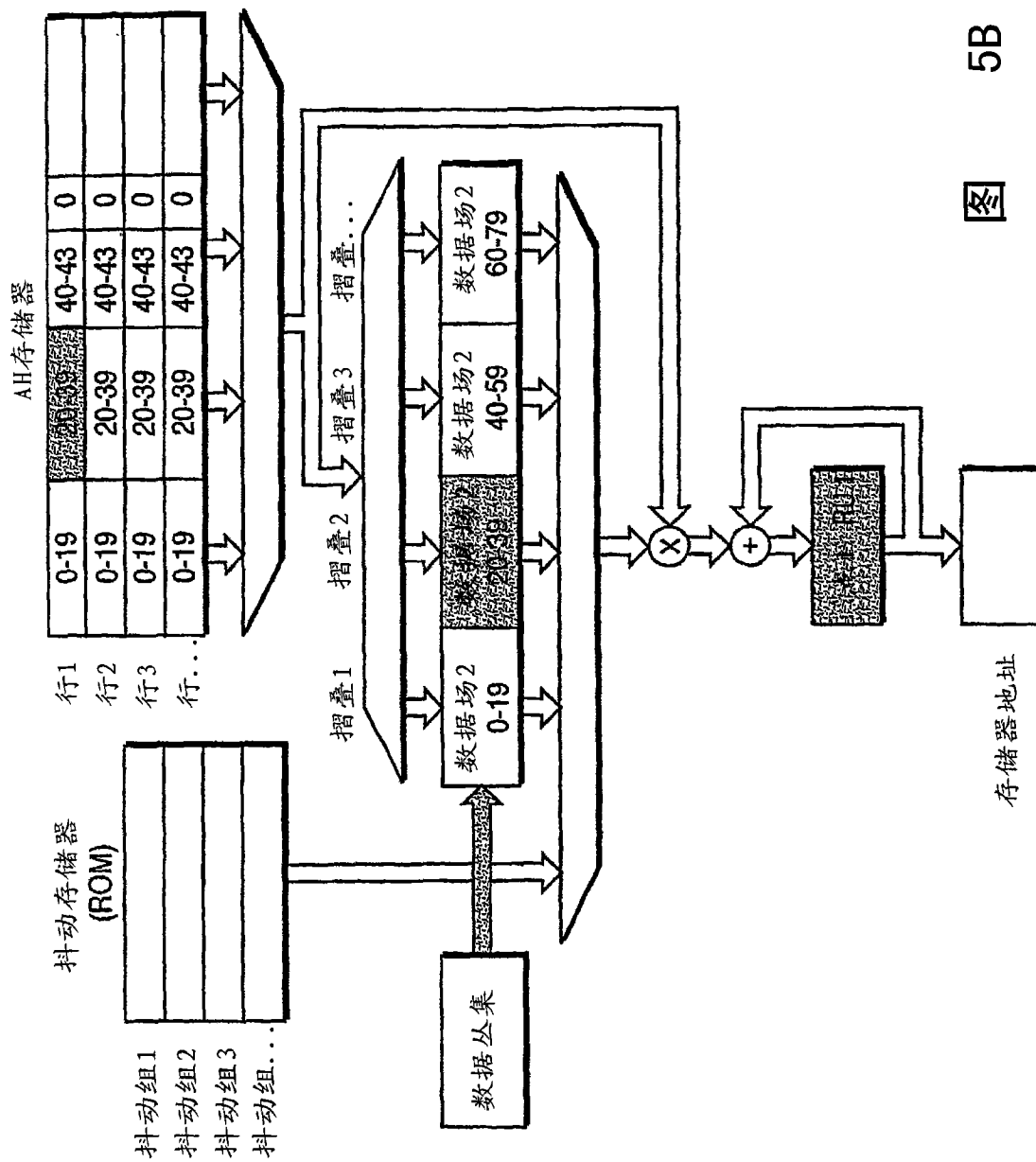
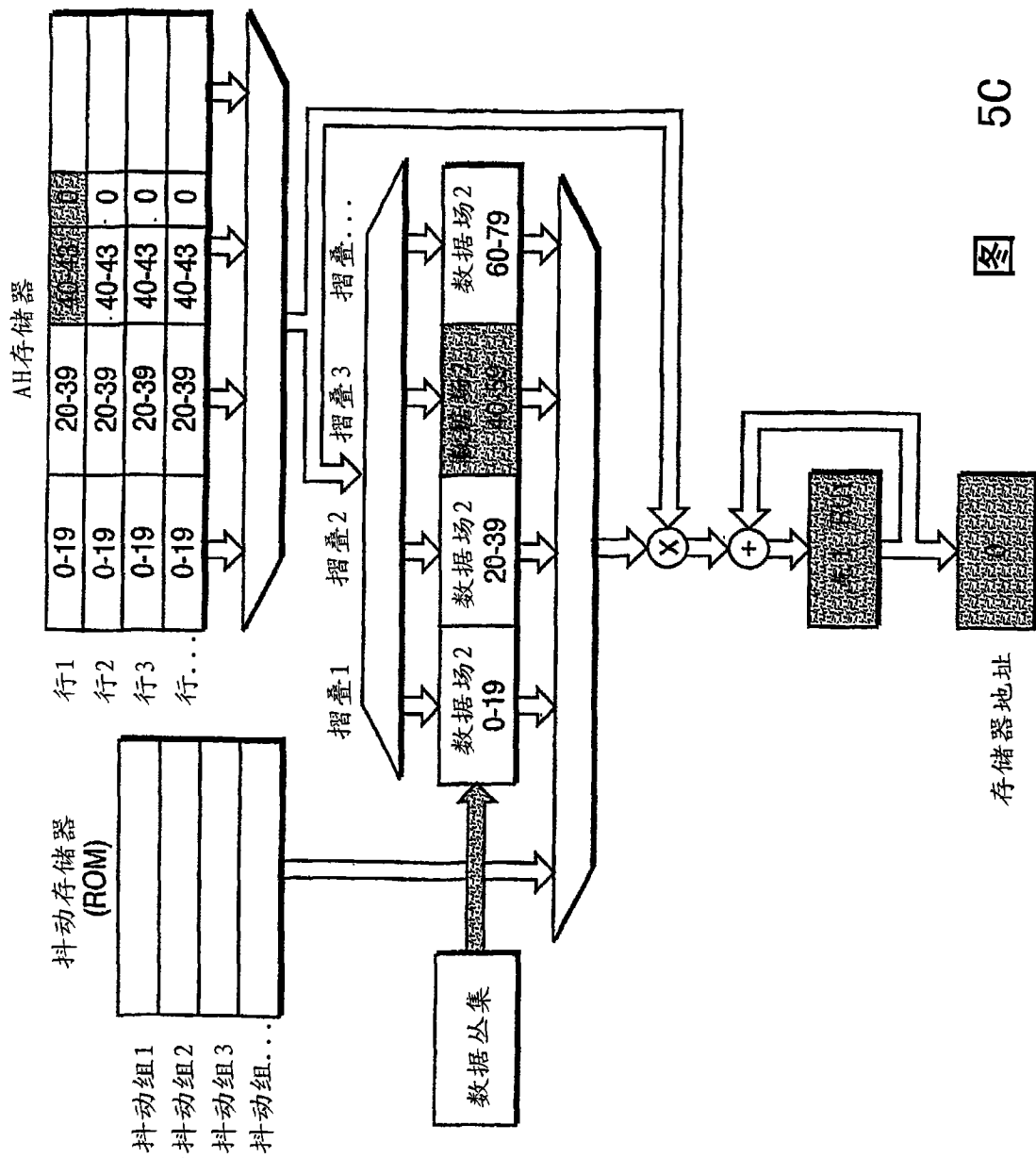


图 5B



50

图

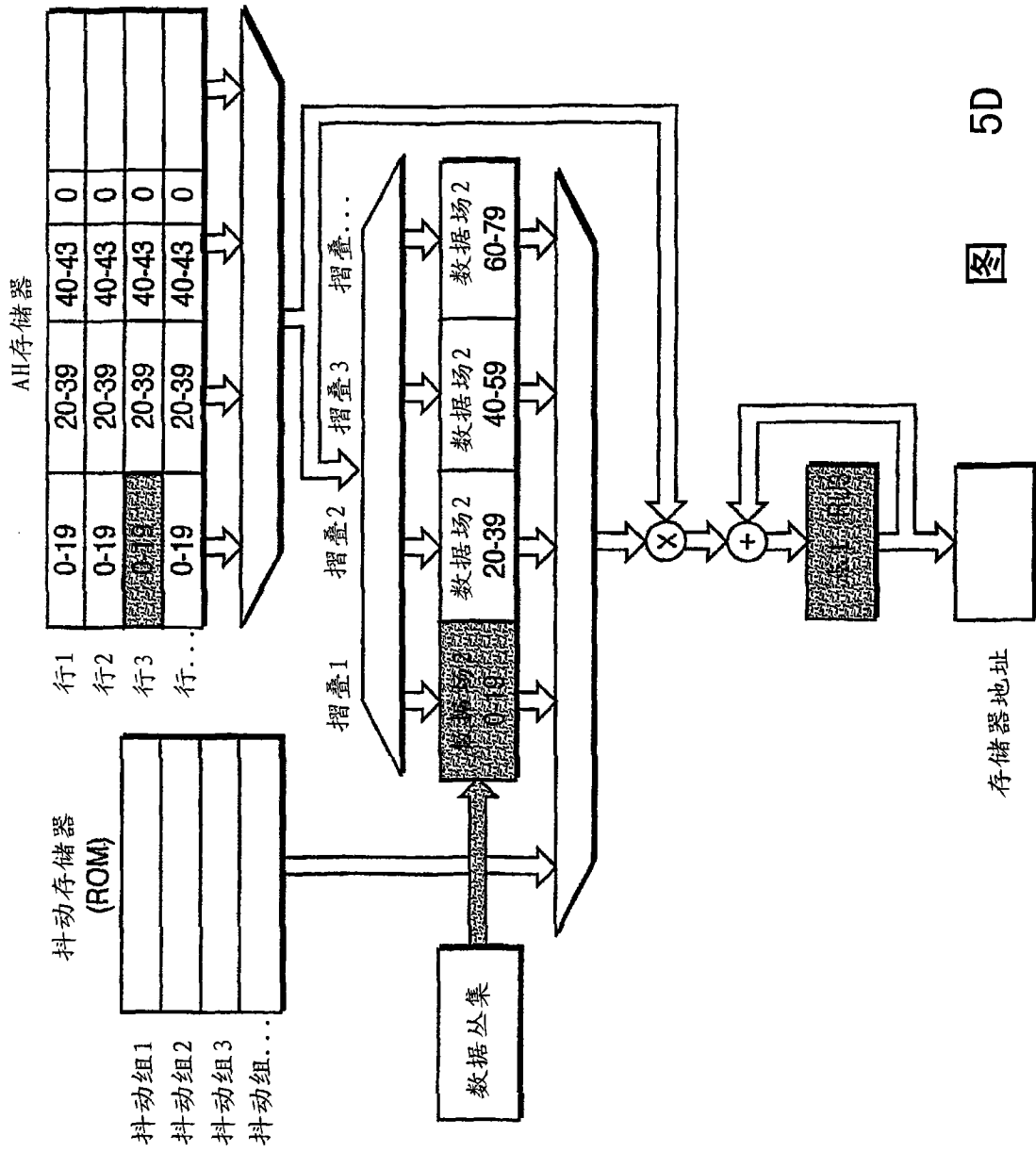


图 5D

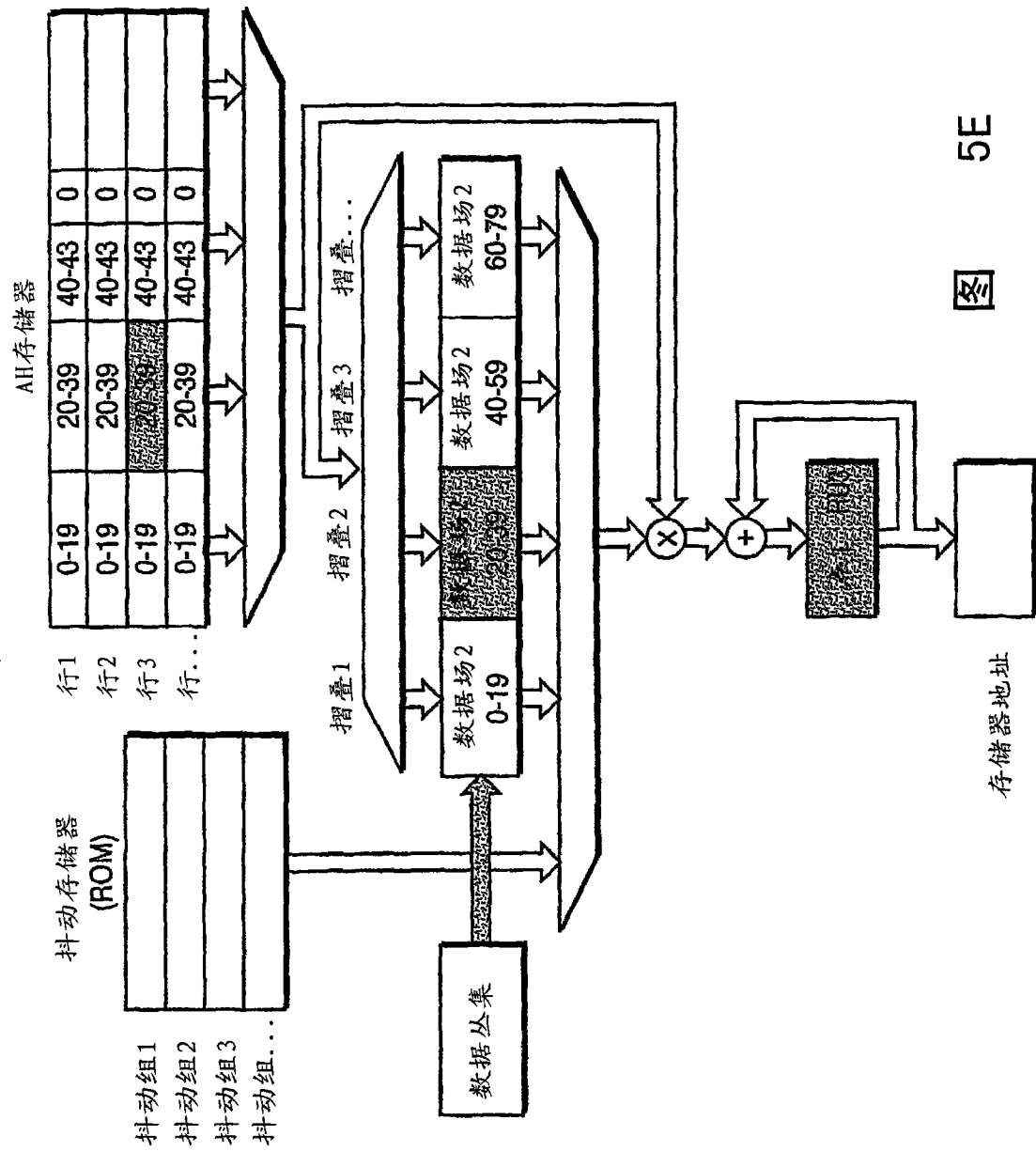


图 5E

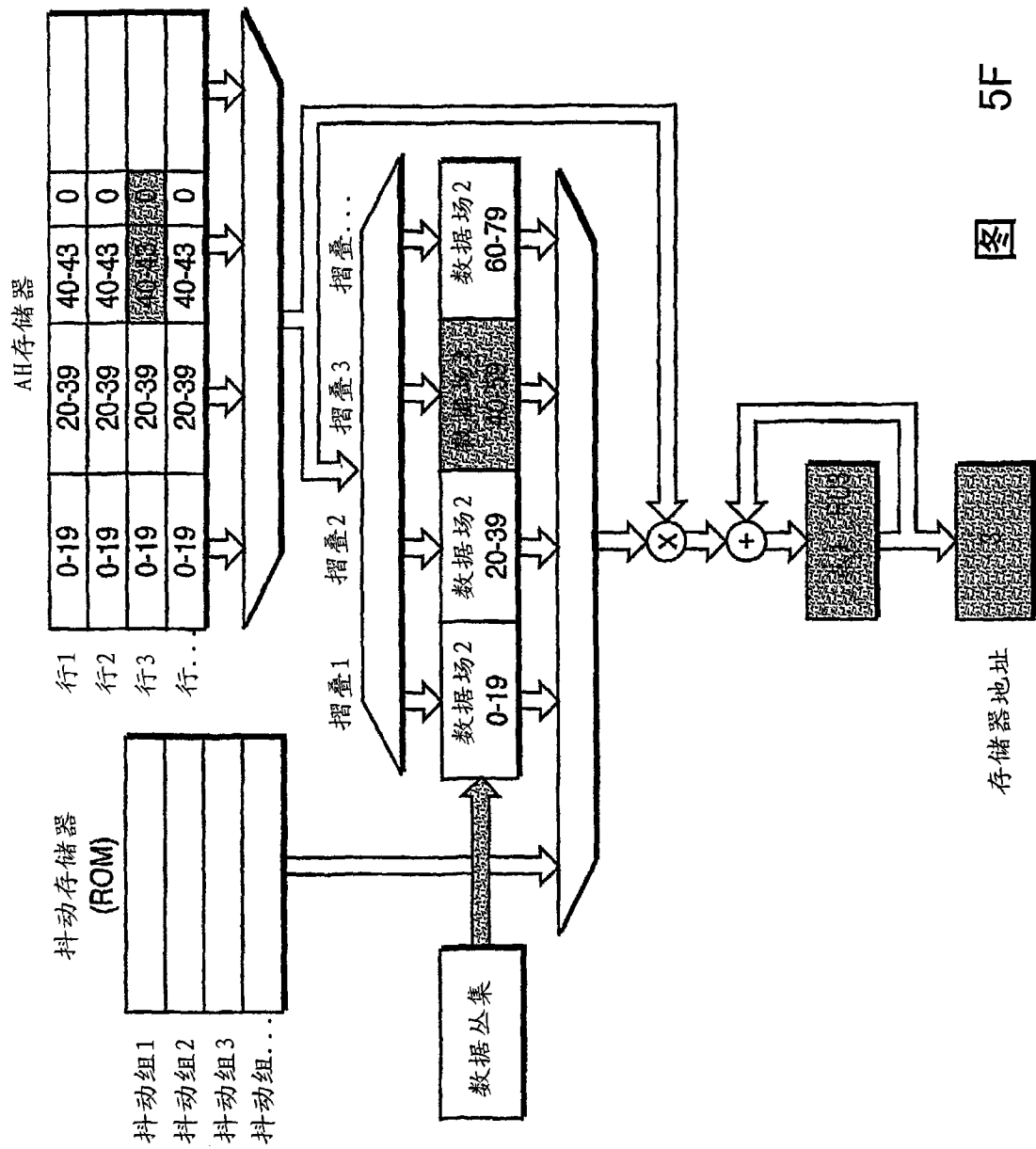
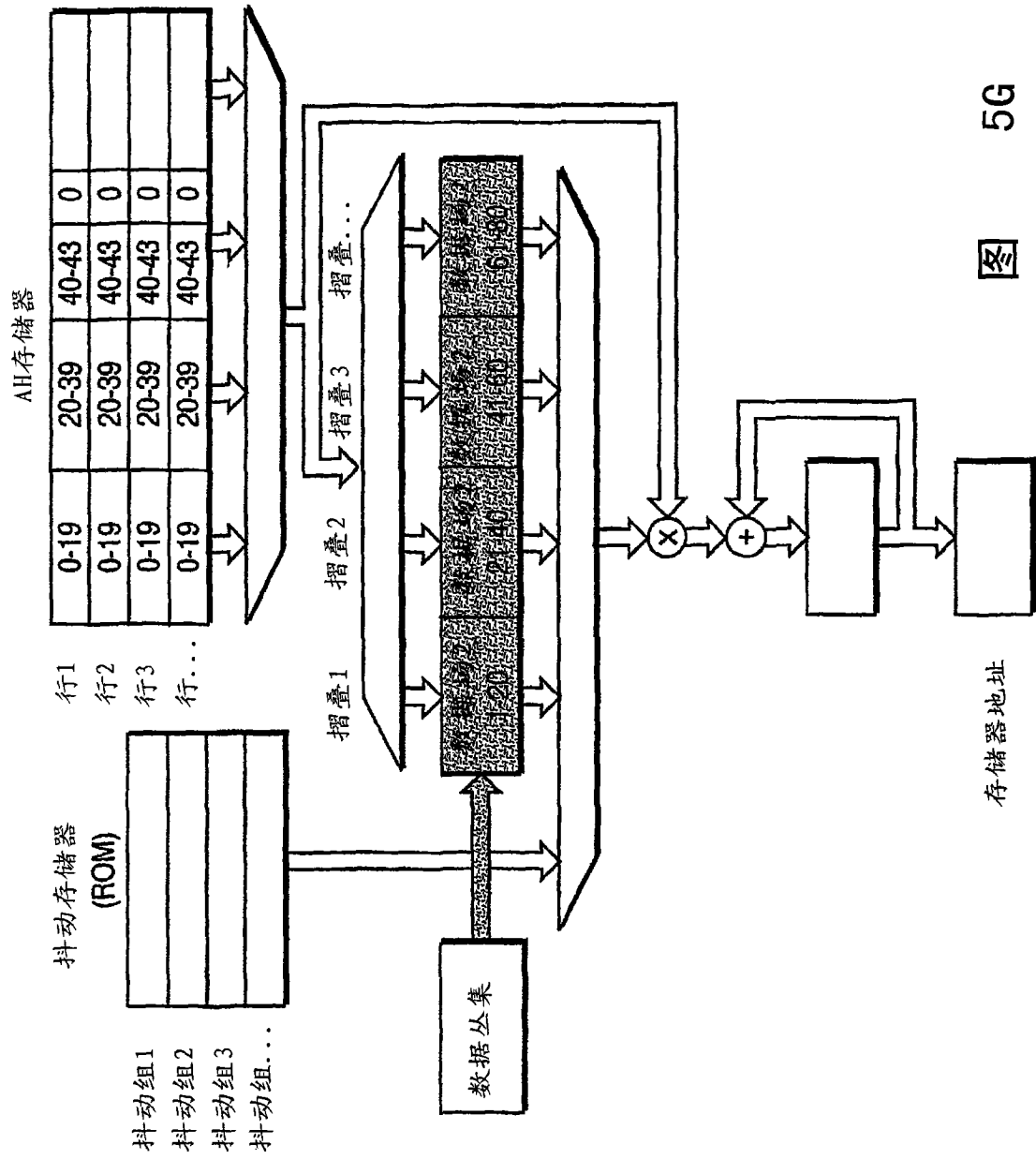
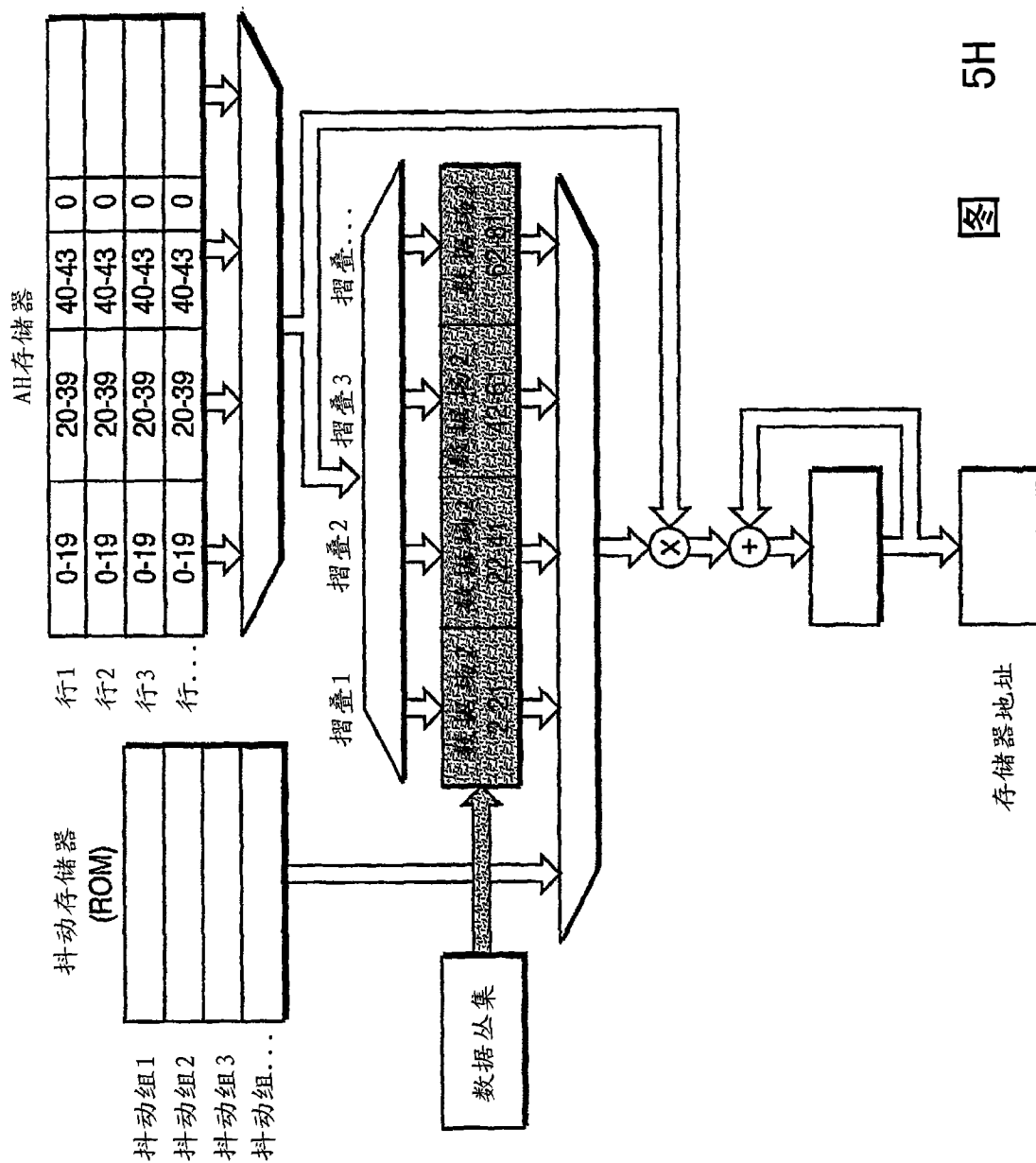
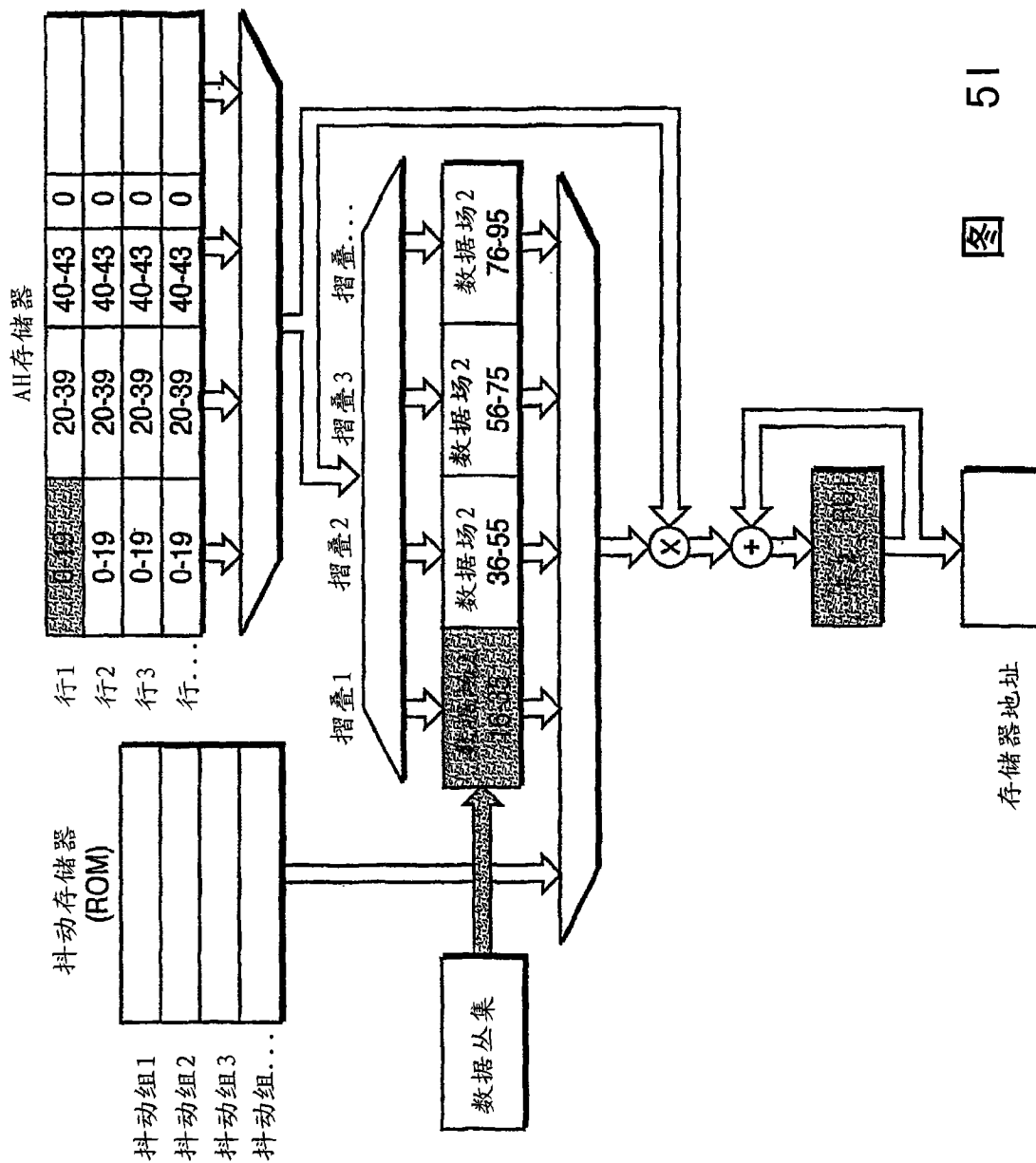


图 5F

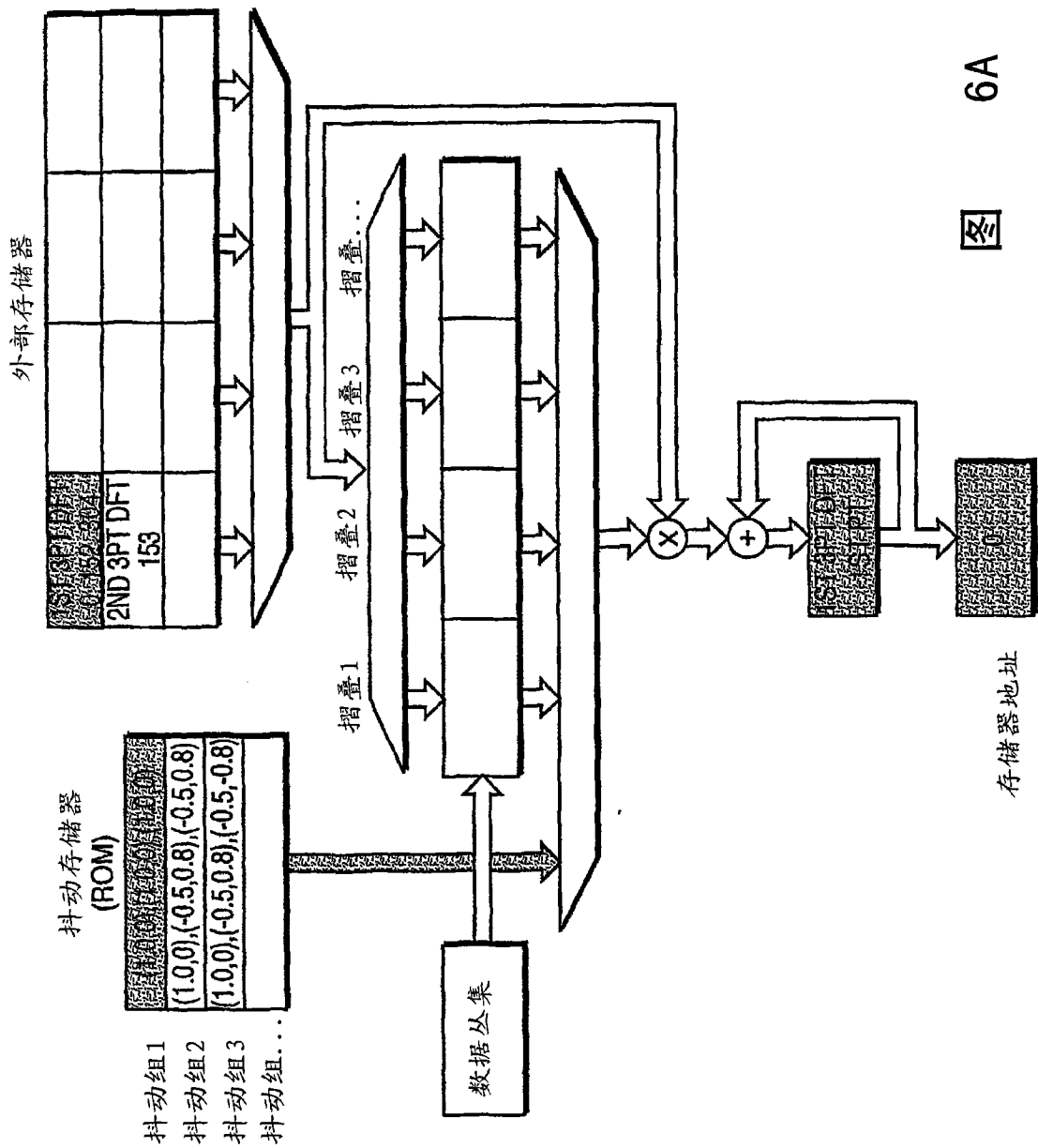






51





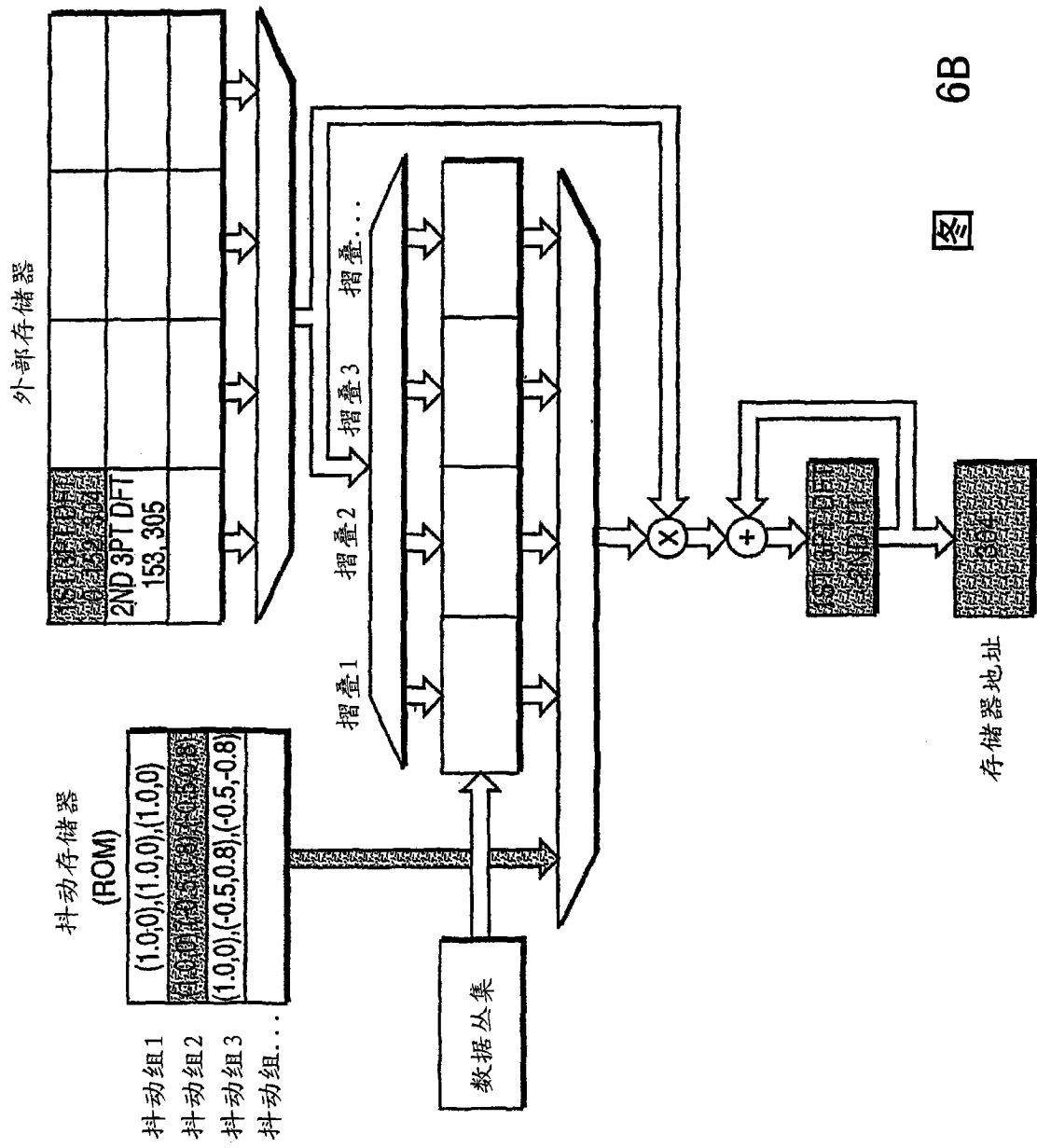
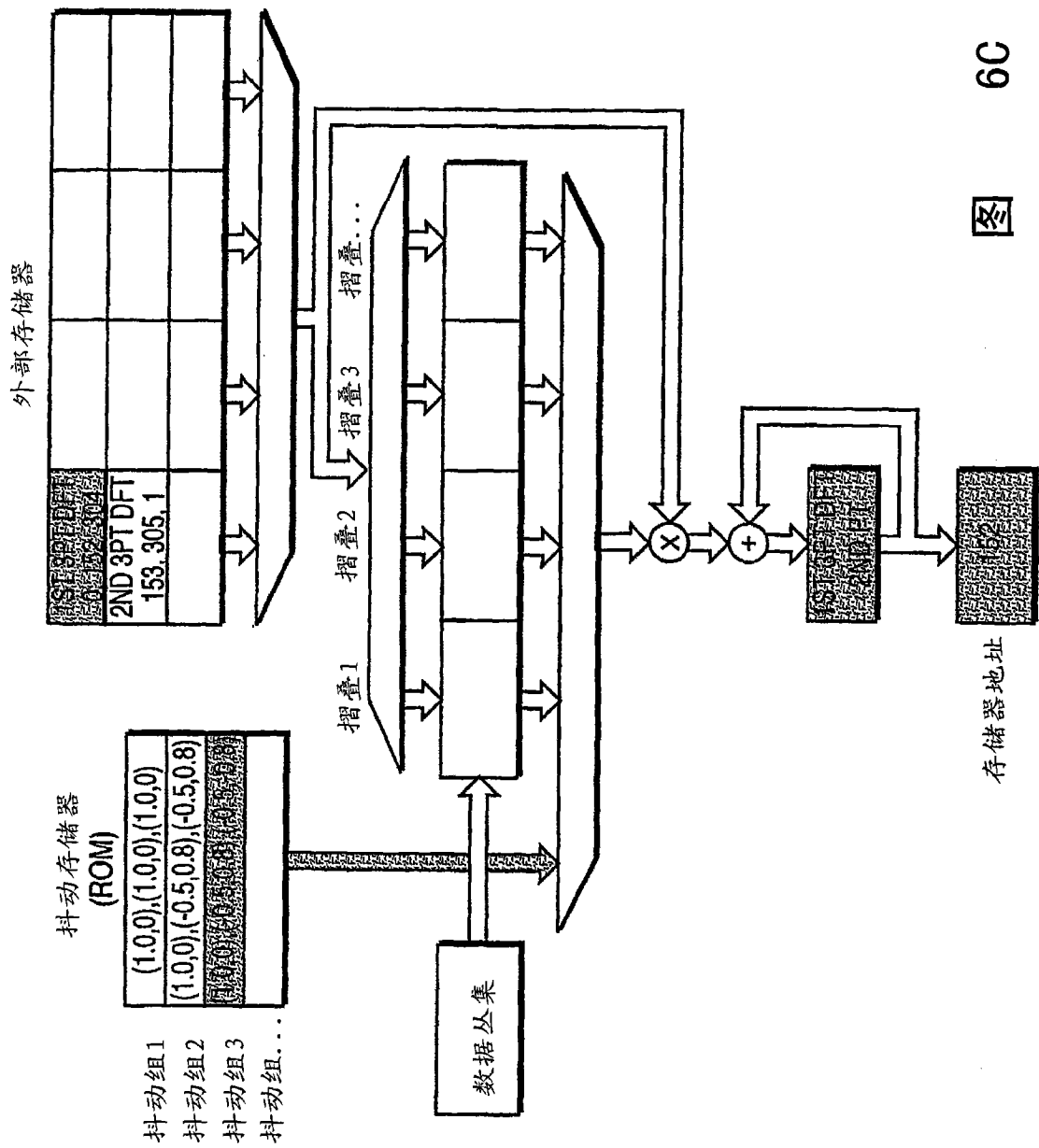


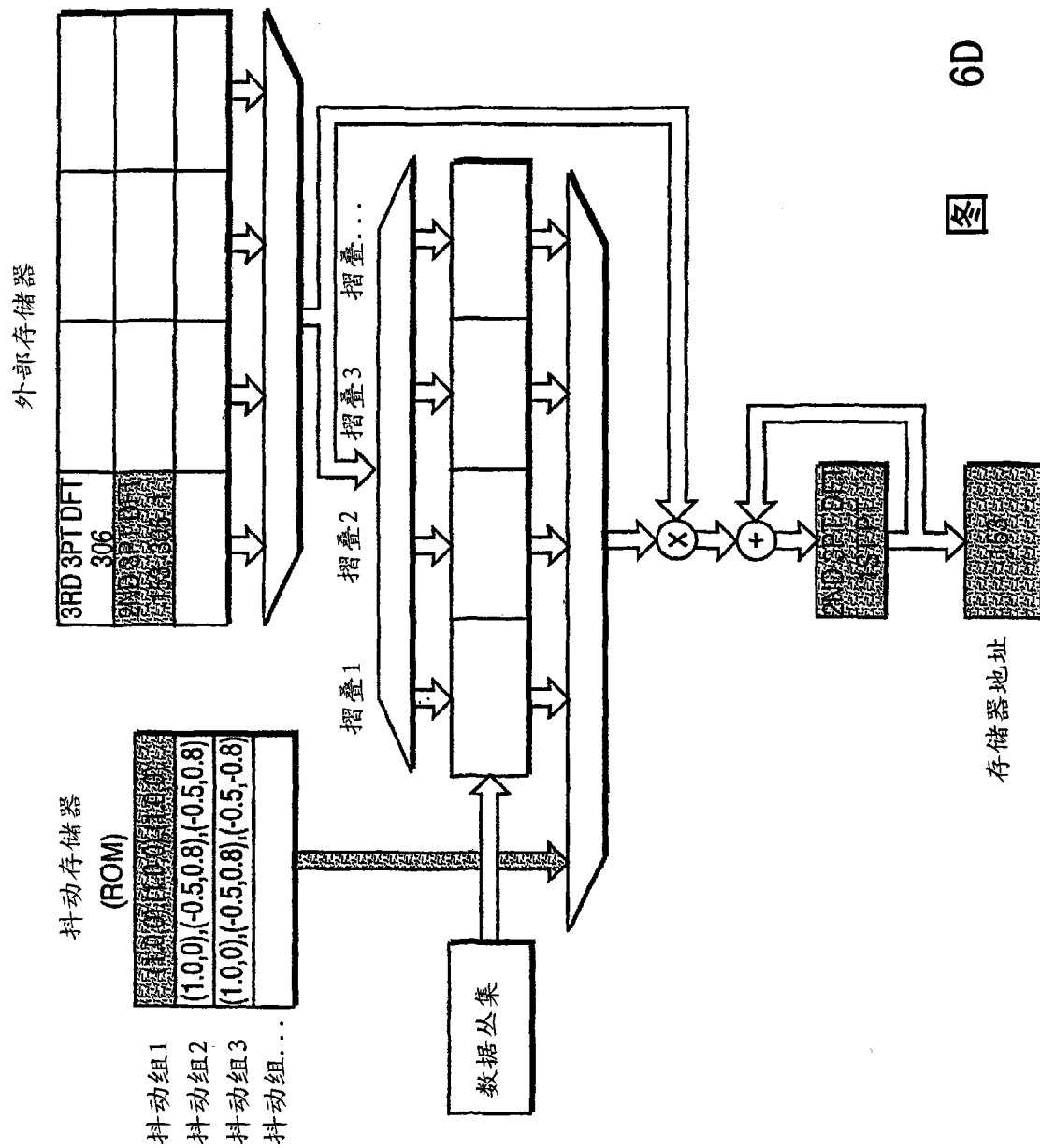
图 6B



6C



存储器地址



图

6D

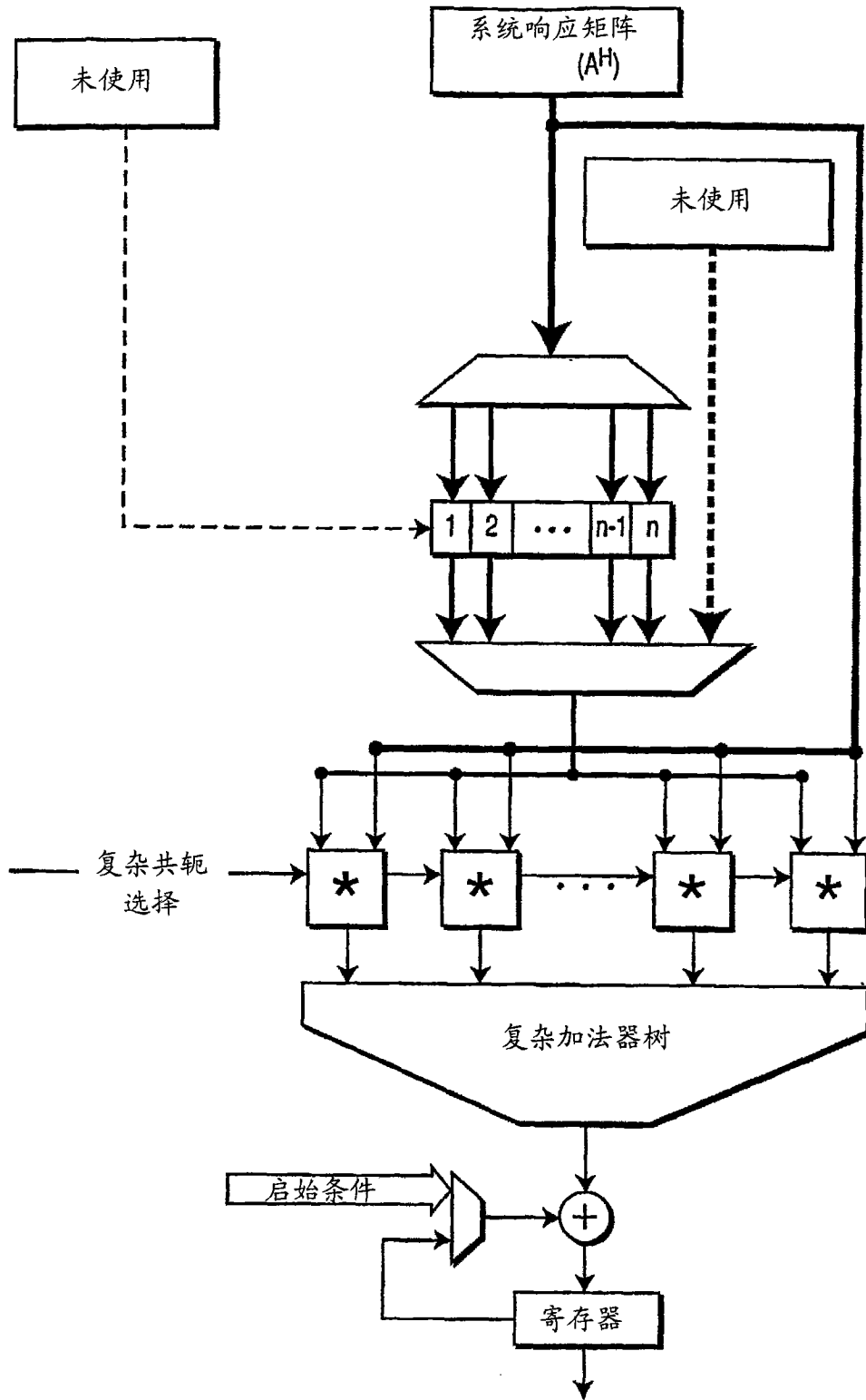


图 7A

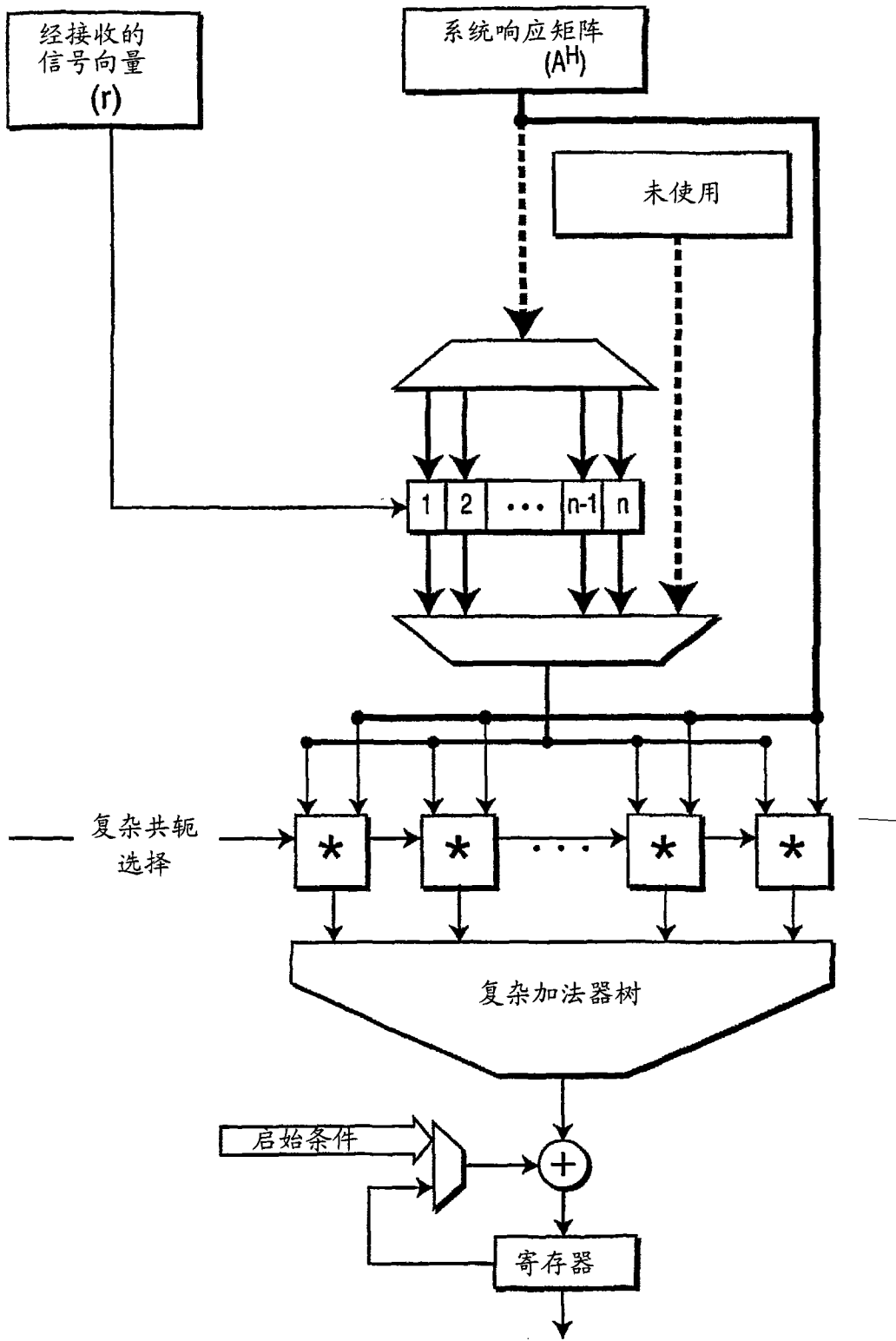


图 7B

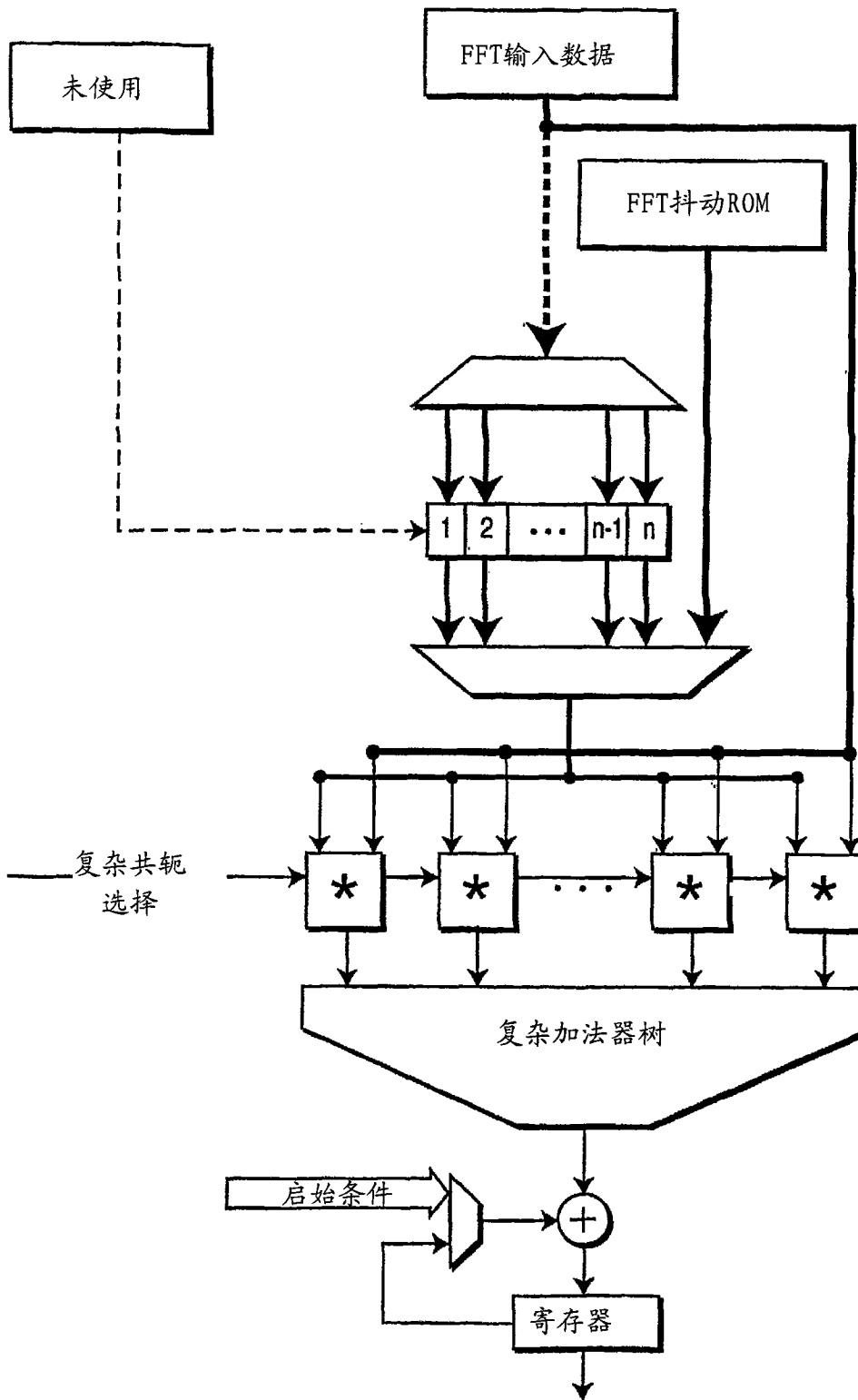


图 7C