

[19] 中华人民共和国国家知识产权局



# [12] 发明专利申请公布说明书

[21] 申请号 200580037481.2

[51] Int. Cl.

G06F 12/02 (2006.01)

G06F 15/00 (2006.01)

G06F 15/76 (2006.01)

G06T 1/00 (2006.01)

G09G 5/36 (2006.01)

G09G 5/39 (2006.01)

[43] 公开日 2009年2月18日

[11] 公开号 CN 101371233A

[22] 申请日 2005.11.14

[21] 申请号 200580037481.2

[30] 优先权

[32] 2004.11.15 [33] US [31] 60/628,414

[32] 2005.11.4 [33] US [31] 11/267,638

[32] 2005.11.4 [33] US [31] 11/267,875

[32] 2005.11.4 [33] US [31] 11/267,599

[32] 2005.11.4 [33] US [31] 11/267,700

[86] 国际申请 PCT/US2005/041329 2005.11.14

[87] 国际公布 WO2006/055546 英 2006.5.26

[85] 进入国家阶段日期 2007.4.29

[71] 申请人 辉达公司

地址 美国加利福尼亚州

[72] 发明人 希里什·加德雷

阿希什·卡兰迪卡尔 斯蒂芬·卢

克里斯托弗·T·程

[74] 专利代理机构 北京律盟知识产权代理有限公司

代理人 王允方 刘国伟

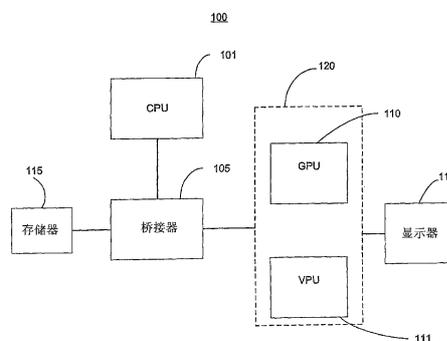
权利要求书6页 说明书42页 附图6页

[54] 发明名称

含控制用于视频处理的向量组件的标量组件的视频处理器

[57] 摘要

本发明描述一种用于执行视频处理操作的延迟容许系统。还描述视频处理器中的一种流处理，所述视频处理器具有标量和向量组件；和在视频处理器中的多维数据路径处理。



1. 一种系统，其包括：
  - 标量执行单元，其经配置以执行标量视频处理操作；
  - 向量执行单元，其经配置以执行向量视频处理操作；
  - 数据存储存储器，其用于存储所述向量执行单元的数据，其中所述数据存储存储器包括多个瓷砖片，所述瓷砖片包括以阵列形式布置的对称的库数据结构，且其中所述库数据结构经配置以支持对每个库的不同瓷砖片的存取。
2. 根据权利要求 1 所述的系统，其中所述系统是所述视频处理器的多维数据路径处理系统，其用于执行视频处理操作。
3. 一种用于进行多维数据路径处理以支持视频处理操作的系统，其包括：
  - 母板；
  - 主机 CPU，其耦接到所述母板；
  - 视频处理器，其耦接到所述母板且耦接到所述 CPU 并包括根据权利要求 1 所述的系统。
4. 根据权利要求 1、2 或 3 所述的系统，其中所述库数据结构的每一者均包含以 4×4 图案布置的多个瓷砖片。
5. 根据权利要求 1、2 或 3 所述的系统，其中所述库数据结构的每一者均包含以 8×8、8×16 或 16×24 图案布置的多个瓷砖片。
6. 根据权利要求 1、2 或 3 所述的系统，其中所述库数据结构经配置以支持对每个库数据结构的多个不同瓷砖片的存取，且其中至少一个存取是针对包括两个相邻库数据结构的瓷砖片的行的所述两个相邻库数据结构。
7. 根据权利要求 1、2 或 3 所述的系统，其中所述瓷砖片经配置以支持对每个库数据结构的多个不同瓷砖片的存取，且其中至少一个存取是针对包括两个相邻库数据结构的瓷砖片的列的所述两个相邻库数据结构。

8. 根据权利要求 1、2 或 3 所述的系统，其进一步包括：  
交叉矩阵，其耦接到所述数据存储器并用于选择用于存取所述多个库数据结构的瓷砖片的配置。
9. 根据权利要求 8 所述的系统，其中所述交叉矩阵存取所述多个库数据结构的瓷砖片，以便以每时钟为基础向向量数据路径供应数据。
10. 根据权利要求 9 所述的系统，其进一步包括采集器，所述采集器用于接收由所述交叉矩阵存取的所述多个库数据结构的瓷砖片，并以每时钟为基础向所述向量数据路径的前端提供所述瓷砖片。
11. 一种用于执行视频处理操作的视频处理器，其包括：  
主机接口，其用于在所述视频处理器与主机 CPU 之间实施通信；  
存储器接口，其用于在所述视频处理器与帧缓冲存储器之间实施通信；  
标量执行单元，其耦接到所述主机接口和所述存储器接口，且经配置以执行标量视频处理操作；和  
向量执行单元，其耦接到所述主机接口和所述存储器接口，且经配置以执行向量视频处理操作。
12. 一种用于执行视频处理操作的系统，其包括：  
母板；  
主机 CPU，其耦接到所述母板；  
根据权利要求 11 所述的所述视频处理器，其耦接到所述母板且耦接到所述 CPU。
13. 根据权利要求 11 所述的视频处理器，其中所述标量执行单元充当所述视频处理器的控制器并控制所述向量执行单元的操作。
14. 根据权利要求 11 所述的视频处理器，其进一步包括向量接口单元，所述向量接口单元用于介接所述标量执行单元与所述向量执行单元。
15. 根据权利要求 11 所述的视频处理器，其中所述标量执行单元和所述向量执行单元经

配置以异步操作。

16. 根据权利要求 15 所述的视频处理器或根据权利要求 12 所述的系统，其中所述标量执行单元以第一时钟频率执行，且所述向量执行单元以第二时钟频率执行。
17. 根据权利要求 11 所述的视频处理器或根据权利要求 12 所述的系统，其中所述标量执行单元经配置以执行应用程序的流量控制算法，且所述向量执行单元经配置以执行所述应用程序的像素处理操作。
18. 根据权利要求 17 所述的视频处理器，其中所述向量执行单元经配置以在需求驱动基础上在所述标量执行单元的控制下操作。
19. 根据权利要求 17 所述的视频处理器或系统，其中所述标量执行单元经配置以使用命令 FIFO 向所述向量执行单元发送函数调用，且其中所述向量执行单元在需求驱动基础上通过存取所述命令 FIFO 来操作。
20. 根据权利要求 17 所述的视频处理器或系统，其中所述视频处理器的异步操作经配置以支持所述应用程序的向量程序或标量程序的分离的独立更新。
21. 根据权利要求 11 所述的视频处理器，其中所述标量执行单元经配置以使用 VLIW(超长指令字) 代码来操作。
22. 一种用于视频处理器的基于流的存储器存取系统，其用于执行视频处理操作，所述系统包括：
  - 标量执行单元，其经配置以执行标量视频处理操作；
  - 向量执行单元，其经配置以执行向量视频处理操作；
  - 帧缓冲存储器，其用于存储所述标量执行单元和所述向量执行单元的数据；和
  - 存储器接口，其用于在所述标量执行单元和所述向量执行单元与所述帧缓冲存储器之间实施通信，其中所述帧缓冲存储器包括多个瓷砖片，且其中所述存储器接口针对所述向量执行单元或所述标量执行单元实施包括瓷砖片的第一顺序存取的第一流和实施包括瓷砖片的第二顺序存取的第二流。

23. 一种用于执行基于流的存储器存取以支持视频处理操作的系统，其包括：  
    母板；  
    主机 CPU，其耦接到所述母板；  
    视频处理器，其耦接到所述母板且耦接到所述 CPU，所述视频处理器包括：  
        主机接口，其用于在所述视频处理器与所述主机 CPU 之间建立通信；  
        标量执行单元，其耦接到所述主机接口且经配置以执行标量视频处理操作；  
        向量执行单元，其耦接到所述主机接口且经配置以执行向量视频处理操作；和  
        存储器接口，其耦接到所述标量执行单元和所述向量执行单元，且用于在所述标量执行单元和所述向量执行单元与帧缓冲存储器之间建立基于流的通信，其中所述帧缓冲存储器包括多个瓷砖片，且其中所述存储器接口针对所述向量执行单元或所述标量执行单元实施包括瓷砖片的第一顺序存取的第一流和实施包括瓷砖片的第二顺序存取的第二流。
24. 根据权利要求 22 所述的系统，其中所述第一流和所述第二流包括至少一个预取瓷砖片。
25. 根据权利要求 22 所述的系统，其中所述第一流源自所述帧缓冲存储器中的第一位置，且所述第二流源自所述帧缓冲存储器中的第二位置。
26. 根据权利要求 22 或 23 所述的系统，其中所述存储器接口经配置以管理来自多个不同起始位置并到达多个不同终止位置的多个流。
27. 根据权利要求 26 所述的系统，其中所述起始位置中的至少一者或所述终止位置中的至少一者是在系统存储器中。
28. 根据权利要求 22 或 23 所述的系统，其进一步包括：  
    DMA 引擎，其建立到所述存储器接口中且经配置以实施多次存储器读取以支持所述第一流和所述第二流，且实施多次存储器写入以支持所述第一流和所述第二流。
29. 根据权利要求 22 或 23 所述的系统，其中所述第一流经历比所述第二流高的延迟量，

且其中所述第一流为对瓷砖片进行存储并入的缓冲器数量大于所述第二流。

30. 根据权利要求 22 或 23 所述的系统，其中所述存储器接口经配置以预取所述第一流或所述第二流的可调节数目的瓷砖片，以补偿所述第一流或所述第二流的延迟。
31. 一种系统，其包括：
  - 主机接口，其用于在所述视频处理器与所述主机 CPU 之间实施通信；
  - 标量执行单元，其耦接到所述主机接口且经配置以执行标量视频处理操作；
  - 向量执行单元，其耦接到所述主机接口且经配置以执行向量视频处理操作；
  - 命令 FIFO，其使所述向量执行单元能够在需求驱动基础上通过存取所述存储器命令 FIFO 来操作；
  - 存储器接口，其用于在所述视频处理器与帧缓冲存储器之间实施通信；和
  - DMA 引擎，其建立到所述存储器接口中，以用于在多个不同存储器位置之间实施 DMA 传递，并用于向数据存储存储器和指令高速缓冲存储器中加载用于所述向量执行单元的数据和指令。
32. 根据权利要求 31 所述的系统，其中所述系统是用于执行视频处理操作的延迟容许系统。
33. 根据权利要求 32 所述的系统，其进一步包括：
  - 主板；
  - 主机 CPU，其耦接到所述主板；
  - 视频处理器，其耦接到所述主板且耦接到所述 CPU。
34. 根据权利要求 31、32 或 33 所述的系统，其中所述向量执行单元经配置以通过存取所述命令 FIFO 以在所述需求驱动基础上操作而相对于所述标量执行单元异步地操作。
35. 根据权利要求 31、32 或 33 所述的系统，其中所述需求驱动基础经配置以隐藏从所述不同存储器位置到所述向量执行单元的命令 FIFO 的数据传递的延迟。

- 
36. 根据权利要求 31、32 或 33 所述的系统，其中所述标量执行单元经配置以实施算法流量控制处理，且其中所述向量执行单元经配置以实施大多数视频处理工作负载。
  37. 根据权利要求 36 所述的系统，其中所述标量执行单元经配置以预计算所述向量执行单元的工作参数以隐藏数据传递延迟。
  38. 根据权利要求 31 所述的系统，其中所述向量执行单元经配置以经由所述 DMA 引擎调度存储器读取以预取用于随后执行向量子程序的命令。
  39. 根据权利要求 38 所述的系统，其中所述存储器读取经调度以在由所述标量执行单元对所述向量子程序进行调用之前预取用于执行所述向量子程序的命令。
  40. 根据权利要求 33 所述的系统，其中所述向量处理单元经配置以经由所述 DMA 引擎调度存储器读取以预取用于随后执行向量子程序的命令，且其中所述存储器读取经调度以在由所述标量执行单元对所述向量子程序进行调用之前预取用于执行所述向量子程序的命令。

## 含控制用于视频处理的向量组件的标量组件的视频处理器

本申请案主张基于 Gadre 等人的于 2004 年 11 月 15 日申请的题为“A METHOD AND SYSTEM FOR VIDEO PROCESSING”的第 60/628,414 号美国临时申请案在 35 U.S.C.第 119 条 (e) 规定下的优先权，所述申请案的全文并入本文中。

### 技术领域

本发明的领域涉及数字电子计算机系统。更明确地说，本发明涉及在计算机系统上有效地处理视信息的系统。一方面描述了用于执行视频处理操作的延迟容许系统。另一方面描述了视频处理器中的流处理。另外描述了视频处理器中的多维数据路径处理。还描述了具有标量和向量组件的视频处理器。

### 背景技术

近年来，图像和全运动视频的显示是电子行业中取得很大进步的领域。高质量视频（尤其是高清晰度数字视频）的显示和再现是现代视频技术应用和装置的主要目标。视频技术用于各种产品，其范围涵盖蜂窝式电话、个人视频录像机、数字视频投影仪、高清晰度电视等。能够产生和显示高清晰度视频的装置的出现和日渐增多的使用是电子行业中历经较大程度的创新和进展的领域。

在许多消费电子产品型装置和专业级装置中使用的视频技术依赖于一个或一个以上视频处理器来格式化和/或增强视频信号以进行显示。这对于数字视频应用来说尤其如此。例如，一个或一个以上视频处理器并入典型机顶盒中，且用于将 HDTV 广播信号转换成显示器可用的视频信号。此种转换涉及（例如）进行缩放，其中视频信号从非 16×9 视频图像进行转换以用于在真正的 16×9（例如，宽屏幕）显示器上正确显示。一个或一个以上视频处理器可用于执行扫描转换，其中将视频信号从隔行格式转换成逐行格式，其中，在隔行格式下奇数和偶数扫描线分离显示，在逐行格式下在单次扫描中提取整个帧。

视频处理器应用的额外实例包含（例如）信号解压缩，其中视频信号以压缩格式（如，MPEG-2）接收且经解压缩和格式化以进行显示。另一实例是重新进行隔行扫描转换，这涉及将进入的数字视频信号从 DVI（数字视觉接口）格式转换成复合视频格式，所述复

合视频格式与市面上安装的大量旧电视显示器兼容。

较高级的用户需要较高级的视频处理器功能，例如（举例来说）回路内/回路外解块滤波、高级的运动自适应去隔行、用于编码操作的输入噪声滤波、多相缩放/重新取样、子图片复合和处理器-放大器操作（例如色空间转换、调整、像素点操作（如，锐化、直方图调整等）和各种视频表面（video surface）格式转换支持操作）。

关于提供此高级视频处理器功能的问题是：具有足以实施此功能的强大架构的视频处理器要并入到许多类型的装置中可能是非常昂贵的。视频处理功能越高级，实施这类功能所需要的集成电路装置在硅片（silicon die）面积、晶体管数、存储器速度要求等方面就越昂贵。

因此，现有技术系统的设计者被迫在视频处理器性能和成本方面作出折中。广泛认为具有可接受的成本/性能比的现有技术视频处理器在延迟约束（如，为避免使视频出现停顿或以其它方式使视频处理程序停止）和计算密度（如，每平方毫米电路小片的处理器操作数目）方面通常很难满足。此外，现有技术视频处理器通常不适于线性缩放性能要求，例如，在期望视频装置处理多个视频流的情况下（如，同时处理多个进入流和外出显示流）。

因此，需要克服现有技术上的限制的新视频处理器系统。新的视频处理器系统应该可调整且具有较高的计算密度，以处理日益高级的用户所期望的高级视频处理器功能。

## **发明内容**

本发明的实施例提供一种新视频处理器系统，其支持高级的视频处理功能，同时有效利用集成电路硅片面积、晶体管数、存储器速度要求等。本发明的实施例保持高计算密度，并易于调整以处理多个视频流。

在一个实施例中，实施一种用于在视频处理器中执行视频处理操作的等待容许系统。所述系统包含：主机接口，其用于实施视频处理器与主机 CPU 之间的通信；标量执行单元，其耦接到主机接口并经配置以执行标量视频处理操作；和向量执行单元，其耦接到主机接口并经配置以执行向量视频处理操作。包含命令 FIFO 以使向量执行单元能够在需求驱动基础（demand driven basis）上通过存取存储器命令 FIFO 来操作。包含存储器接口以在视频处理器与帧缓冲存储器之间实施通信。在存储器接口中建立 DMA 引擎以在多个不同存储器位置之间实施 DMA 传递，并向数据存储存储器和指令高速缓冲存储器中加载用于向量执行单元的数据和指令。

在一个实施例中，向量执行单元经配置通过存取命令 FIFO 而相对于标量执行单元异步地操作以在需求驱动基础上操作。需求驱动基础可经配置以隐藏从不同存储器位置（如，帧缓冲存储器、系统存储器、高速缓冲存储器等）到向量执行单元的命令 FIFO 的数据传递的延迟。命令 FIFO 可为管线 FIFO 以防止向量执行单元停止。

在一个实施例中，将本发明实施为用于执行视频处理操作的视频处理器。所述视频处理器包含主机接口，其用于在视频处理器与主机 CPU 之间实施通信。所述视频处理器包含存储器接口，其用于在视频处理器与帧缓冲存储器之间执行通信。标量执行单元耦接到主机接口和存储器接口，且经配置以执行标量视频处理操作。向量执行单元耦接到主机接口和存储器接口，且经配置以执行向量视频处理操作。视频处理器可为独立视频处理器集成电路或可为集成到 GPU 集成电路的组件。

在一个实施例中，标量执行单元充当视频处理器的控制器并控制向量执行单元的操作。标量执行单元可经配置以执行应用程序的流量控制算法，且向量执行单元可经配置以执行应用程序的像素处理操作。向量接口单元可包含在视频处理器中，以介接标量执行单元与向量执行单元。在一个实施例中，标量执行单元和向量执行单元经配置以异步操作。标量执行单元可以第一时钟频率执行且向量执行单元可以不同时钟频率（如，较快、较慢等）执行。向量执行单元可在需求驱动基础上在标量执行单元的控制下操作。

在一个实施例中，本发明实施为视频处理器的多维数据路径处理系统以用于执行视频处理操作。所述视频处理器包含标量执行单元，其经配置以执行标量视频处理操作；和向量执行单元，其经配置以执行向量视频处理操作。包含数据存储存储器以存储向量执行单元的数据。数据存储存储器包含多个瓷砖片（tile），所述瓷砖片具有以阵列形式布置的对称库数据结构。所述库数据结构经配置以支持对每个库的不同瓷砖片的存取。

根据特定配置的要求，每个库数据结构可包括多个瓷砖片（如，4×4、8×8、8×16、16×24 等）。在一个实施例中，所述库可经配置以支持对每个库的不同瓷砖片的存取。这允许单个存取以从两个相邻的库中检索瓷砖片的行或列。在一个实施例中，使用交叉矩阵（crossbar）来选择用于存取多个库数据结构（如，行、列、块等）的瓷砖片的配置。可包含采集器以用于接收交叉矩阵所存取的库的瓷砖片，并以每时钟为基础向向量数据路径的前端提供瓷砖片。

在一个实施例中，本发明实施为视频处理器的基于流的存储器存取系统。所述视频处理器包含标量执行单元，其经配置以执行标量视频处理操作；和向量执行单元，其经配置以执行向量视频处理操作。包含帧缓冲存储器以用于存储标量执行单元和向量执行

单元的数据。包含存储器接口以在标量执行单元和向量执行单元与帧缓冲存储器之间执行通信。帧缓冲存储器包括多个瓷砖片。存储器接口针对标量执行单元实施瓷砖片的第一顺序存取的第一流，且针对向量执行单元实施瓷砖片的第二顺序存取的第二流。

在一个实施例中，第一流和第二流包括一系列顺序预取的瓷砖片，所述瓷砖片是以隐藏来自起始存储器位置（如，帧缓冲存储器、系统存储器等）的存取延迟的方式预取的。在一个实施例中，存储器接口经配置以管理来自多个不同起始位置且到达多个不同终止位置的多个不同流。在一个实施例中，使用建立在存储器接口中的 DMA 引擎来实施多次存储器读取和多次存储器写入以支持多个流。

概括地说，本发明揭示至少如下四种方法。

A)在此描述内容中概括地讲述的方法是视频处理器中的多维数据路径处理系统执行视频处理操作的方法，所述方法包括：通过使用标量执行单元来执行标量视频处理操作；通过使用向量执行单元来执行向量视频处理操作；通过使用数据存储存储器来存储向量执行单元的数据，其中数据存储存储器包括多个瓷砖片，所述瓷砖片包括以阵列形式布置的对称库数据结构，且其中库数据结构经配置以支持对每个库的不同瓷砖片的存取。另外，上述方法 A 包括包含以 4×4 图案布置的多个瓷砖片的各个库数据结构。并且，上述方法 A 包括包含以 8×8、8×16 或 16×24 图案布置的多个瓷砖片的各个库数据结构。另外，上述方法 A 包括经配置以支持对每个库数据结构的各个不同瓷砖片的存取的库数据结构，其中至少一个存取是存取包括两个库数据结构的瓷砖片的行的两个相邻库数据结构。上述方法 A 还涉及瓷砖片经配置以支持对每个库数据结构的各个不同瓷砖片的存取，其中至少一个存取是存取包括两个相邻库数据结构的瓷砖片的列的两个相邻库数据结构。此外，上述方法 A 包括通过使用耦接到数据存储器的交叉矩阵来选择用于存取多个库数据结构的瓷砖片的配置。在此选择步骤中，交叉矩阵存取多个库数据结构的瓷砖片从而以每时钟为基础向向量数据路径供应数据。并且，此处涉及通过使用采集器来接收交叉矩阵所存取的多个库数据结构的瓷砖片；并以每时钟为基础向向量数据路径的前端提供瓷砖片。

B)在此描述内容中概括地讲述的方法也是用于执行视频处理操作的方法，所述方法使用执行计算机可读代码的计算机系统的视频处理器来实施，其包括：通过使用主机接口在视频处理器与主机 CPU 之间建立通信；通过使用存储器接口在视频处理器与帧缓冲存储器之间建立通信；通过使用耦接到主机接口和存储器接口的标量执行单元来执行标量视频处理操作；和通过使用耦接到主机接口和存储器接口的向量执行单元来执行向量视频处理操作。上述方法 B 进一步包括充当视频处理器的控制器并控制向量执行单元的

操作的标量执行单元。上述方法 B 还包括用于介接标量执行单元与向量执行单元的向量接口单元。上述方法 B 还包括经配置以异步操作的标量执行单元和向量执行单元。并且，标量执行单元以第一时钟频率执行且向量执行单元以第二时钟频率执行。上述方法 B 包括经配置以执行应用程序的流量控制算法的标量执行单元和经配置以执行应用程序的像素处理操作的向量执行单元。另外，向量执行单元经配置以在需求驱动基础上在标量执行单元的控制下操作。另外，标量执行单元经配置以使用存储器命令 FIFO 向向量执行单元发送函数调用，向量执行单元在需求驱动基础上通过存取存储器命令 FIFO 来操作。并且，视频处理器的异步操作经配置以支持应用程序的向量程序或标量程序的分离的独立更新。最后，上述方法 B 包括标量执行单元经配置以使用 VLIW（超长指令字）代码进行操作。

C) 此处描述的方法也概括地讲述了一种在视频处理器中基于流的存储器存取执行视频处理操作的方法，所述方法包括：通过使用标量执行单元来执行标量视频处理操作；通过使用向量执行单元来执行向量视频处理操作；通过使用帧缓冲存储器来存储标量执行单元和向量执行单元的数据；和通过使用存储器接口而在标量执行单元和向量执行单元与帧缓冲存储器之间实施通信，其中所述帧缓冲存储器包括多个瓷砖片，且其中所述存储器接口针对向量执行单元或标量执行单元实施包括瓷砖片的第一顺序存取的第一流和包括瓷砖片的第二顺序存取的第二流。上述方法 C 还使第一流和第二流包含至少一个预取瓷砖片。上述方法 C 进一步包括源自帧缓冲存储器中的第一位置的第一流，和源自帧缓冲存储器中的第二位置的第二流。上述方法 C 还包括存储器接口经配置以管理来自多个不同起始位置并到达多个不同终止位置的多个不同流。在这个方面，所述起始位置中的至少一者或所述终止位置中的至少一者是系统存储器。上述方法 C 还包括通过使用建立在存储器接口中的 DMA 引擎以实施多次存储器读取以支持第一流和第二流；并实施多次存储器写入以支持第一流和第二流。另外，方法 C 包括第一流经历比第二流高的延迟量，其中第一流为对瓷砖片进行存储并入的缓冲器数量大于第二流。方法 C 还包括存储器接口经配置以预取第一流或第二流的可调节数目的瓷砖片从而补偿第一流或第二流的延迟。

D) 此处描述的方法还概括地包含一种用于延迟容许视频处理操作的方法，所述方法包括：通过使用主机接口而在视频处理器与主机 CPU 之间建立通信；通过使用耦接到主机接口的标量执行单元来执行标量视频处理操作；通过使用耦接到主机接口的向量执行单元来执行向量视频处理操作；使向量执行单元能够在需求驱动基础上通过存取存储器

命令 FIFO 而操作；通过使用存储器接口而在视频处理器与帧缓冲存储器之间实施通信；和通过使用 DMA 引擎而在多个不同存储器位置之间实施 DMA 传递，所述 DMA 引擎建立在存储器接口中并经配置以向数据存储存储器和指令高速缓冲存储器中加载用于向量执行单元的数据和指令。上述方法 D 进一步包括向量执行单元经配置以通过存取命令 FIFO 来操作从而相对于标量执行单元异步操作以在需求驱动基础上操作。上述方法 D 还包括需求驱动基础经配置以隐藏从不同存储器位置到向量执行单元的命令 FIFO 的数据传递的延迟。此外，上述方法 D 还包括标量执行单元经配置以实施算法流量控制处理，且其中向量执行单元经配置以实施大多数的视频处理工作负载。在这种情况下，标量执行单元经配置以预计算向量执行单元的工作参数来隐藏数据传递延迟。上述方法 D 包括向量执行单元经配置以经由 DMA 引擎来调度存储器读取以预取用于随后执行向量子程序的命令。此处，存储器读取经调度以在由标量执行单元对向量子程序进行的调用之前预取用于执行向量子程序的命令。

## **附图说明**

在附图的图式中，以举例方式且不以限制方式说明本发明，且其中相同参考数字表示类似元件，在附图中：

图 1 展示根据本发明一个实施例的计算机系统的基本组件的概览图。

图 2 展示描绘根据本发明一个实施例的视频处理器单元的内部组件的图。

图 3 展示根据本发明一个实施例的视频处理器的示范性软件程序的图。

图 4 展示使用视频处理器且根据本发明一个实施例将视频与子图片混合的实例。

图 5 展示描绘根据本发明一个实施例的向量执行的内部组件的图。

图 6 展示描绘根据本发明一个实施例具有对称的瓷砖片阵列的数据存储存储器的布局的图。

## **具体实施方式**

现在将详细参考本发明的优选实施例，其实例在附图中说明。虽然将结合优选实施例描述本发明，但将理解，优选实施例并不意图将本发明限制于这些实施例。相反，本发明旨在涵盖可能包含在由所附权利要求书所界定的本发明的精神和范畴内的替代、修改和等效物。此外，在本发明实施例的下列详细描述中，陈述了许多特定细节以提供对本发明的透彻理解。然而，所属领域的技术人员将认识到，可在没有这些特定细节的情

况下实践本发明。在其它情况下，为了避免混淆本发明实施例的方面，不再详细描述众所周知的方法、程序、组件和电路。

### 符号和命名

下文详细描述的某些部分是在对计算机存储器内的数据位进行的操作的程序、步骤、逻辑块、处理和其它符号表示方面呈现的。这些描述和表示是数据处理领域的技术人员用来最有效地向所属领域的其他技术人员传达其工作的内容的手段。程序、计算机执行的步骤、逻辑块、处理等在此处且大体上设想成达到所要结果的自相一致的步骤或指令序列。所述步骤是那些需要对物理量进行物理操纵的步骤。通常，尽管并非必要，这些量采用能够在计算机系统中存储、传递、组合、比较和以其它方式操纵的电或磁信号的形式。事实证明，将这些信号表示为位、值、元素、符号、字符、术语、数字等有时候比较方便，主要是因为普遍使用的缘故。

然而，应了解，所有这些和类似术语均与适当的物理量相关，且仅仅是应用于这些量的方便的标记。除非通过以下论述显然另有具体规定，否则应了解，贯穿整个发明，使用例如“处理”或“存取”或“执行”或“存储”或“呈现”等的术语的论述是指计算机系统（如，图 1 的计算机系统 100）或类似的电子计算装置的动作和处理，其将计算机系统的寄存器和存储器内的表示为物理（电子）量的数据操纵并变换成其它类似地表示为计算机系统存储器或寄存器或其它此类信息存储、传输或显示装置内的物理量的数据。

### 计算机系统平台：

图 1 展示根本发明一个实施例的计算机系统 100。计算机系统 100 描绘根据本发明实施例的基本计算机系统的组件，其为特定基于硬件的和基于软件的功能提供执行平台。通常，计算机系统 100 包括至少一个 CPU 101、系统存储器 115 以及至少一个图形处理器单元（GPU）110 和一个视频处理器单元（VPU）111。CPU 101 可经由桥接组件 105 耦接到系统存储器 115，或可经由 CPU 101 内部的存储器控制器（未图示）直接耦接到系统存储器 115。桥接组件 105（如，北桥）可支持连接各种 I/O 装置（如，一个或一个以上硬盘驱动器、以太网适配器、CD ROM、DVD 等）的扩展总线。GPU 110 和视频处理器单元 111 耦接到显示器 112。一个或一个以上额外 GPU 可视情况耦接到系统 100 以进一步增加其计算能力。GPU 110 和视频处理器单元 111 经由桥接组件 105 耦接到 CPU 101 和系统存储器 115。系统 100 可实施为（举例来说）具有耦接到专用图形渲染 GPU 110 的强大的通用 CPU 101 的台式计算机系统或服务器计算机系统。在此实施例中，可包含

添加外围总线、专门图形存储器和系统存储器、IO 装置等的组件。同样，系统 100 可实施为手持装置（如，手机等）或机顶盒视频游戏控制台装置，例如（举例来说）从华盛顿州雷蒙德的 Microsoft Corporation 购得的 Xbox®或从日本东京的 Sony Computer Entertainment Corporation 购得的 PlayStation3®。

应了解，GPU 110 可实施为分立组件、经设计以经由连接器（如，AGP 槽、PCI-Express 槽等）耦接到计算机系统 100 的分立图形卡、分立集成电路电路小片（如，直接安装在母板上）或为包含在计算机系统芯片组组件的集成电路电路小片内的集成 GPU（如，集成在桥芯片 105 内）。另外，可包含本地图形存储器以用于 GPU 110 进行高带宽图形数据存储。另外，应了解，GPU 110 和视频处理器单元 111 可集成在同一集成电路电路小片上（如，作为组件 120），或可为以其它方式连接到计算机系统 100 的母板或安装到母板上的单独的分立集成电路组件。

### 本发明的实施例

图 2 展示描绘根据本发明一个实施例的视频处理器单元 111 的内部组件的图。如图 2 所说明，视频处理器单元 111 包含标量执行单元 201、向量执行单元 202、存储器接口 203 和主机接口 204。

在图 2 实施例中，视频处理器单元（下文简称视频处理器）111 包含用于执行视频处理操作的功能组件。视频处理器 111 使用主机接口 204 来经由桥接器 105 在视频处理器 111 与主机 CPU 101 之间建立通信。视频处理器 111 使用存储器接口 203 在视频处理器 111 与帧缓冲存储器 205 之间建立通信（如，用于耦接的显示器 112，未图示）。标量执行单元 201 耦接到主机接口 204 和存储器接口 203，且经配置以执行标量视频处理操作。向量执行单元耦接到主机接口 204 和存储器接口 203，且经配置以执行向量视频处理操作。

图 2 实施例说明所述视频处理器 111 将其执行功能划分成标量操作和向量操作的方式。标量操作由标量执行单元 201 实施。向量操作由向量执行单元 202 实施。

在一个实施例中，向量执行单元 202 经配置以充当所述标量执行单元 201 的从协处理器。在此实施例中，标量执行单元管理通过将控制流馈入到向量执行单元 202 并管理向量执行单元 202 的数据输入/输出来管理所述向量执行单元 202 的工作负载。控制流通常包括函数参数、子程序自变量等等。在典型视频处理应用程序中，将在标量执行单元 201 上执行应用程序的处理算法的控制流，而将在向量执行单元 202 实施实际像素/数据处理操作。

仍参看图 2，标量执行单元 201 可实施为并入有基于 RISC 的执行技术的 RISC 型标量执行单元。向量执行单元 202 可实施为 SIMD 机器，其具有（例如）一个或一个以上 SIMD 管线。在 2 SIMD 管线实施例中，例如，每个 SIMD 管线可以 16 个像素宽度数据路径（或更宽）来实施，并因此向向量执行单元 202 提供原始计算能力以形成每时钟高达 32 个像素的结果数据输出。在一个实施例中，标量执行单元 201 包含经配置以使用 VLIW（超长指令字）软件代码来操作的硬件，从而以每时钟为基础优化标量操作的并行执行。

在图 2 实施例中，标量执行单元 201 包含耦接到标量处理器 210 的指令高速缓冲存储器 211 和数据高速缓冲存储器 212。高速缓冲存储器 211-212 与存储器接口 203 介接以用于存取外部存储器，例如（举例来说）帧缓冲器 205。标量执行单元 201 进一步包含向量接口单元 213 以与向量执行单元 202 建立通信。在一个实施例中，向量接口单元 213 可包含一个或一个以上同步信箱区（mailbox）214，所述信箱区 214 经配置以实现标量执行单元 201 与向量执行单元 202 之间的异步通信。

在图 2 实施例中，向量执行单元 202 包含向量控制单元 220，其经配置以控制向量执行数据路径、向量数据路径 221 的操作。向量控制单元 220 包含命令 FIFO 225 以从标量执行单元 201 接收指令和数据。指令高速缓冲存储器 222 经耦接以向向量控制单元 220 提供指令。数据存储存储器 223 经耦接以向向量数据路径 221 提供输入数据，并从向量数据路径 221 接收结果数据。数据存储存储器 223 充当用于向量数据路径 221 的指令高速缓冲存储器和数据 RAM。指令高速缓冲存储器 222 和数据存储器 223 耦接到存储器接口 203，以用于存取外部存储器，例如，帧缓冲器 205。图 2 实施例还展示第二向量数据路径 231 和相应的第二数据存储存储器 223（如，虚线轮廓）。应理解，展示第二向量数据路径 231 和第二数据存储存储器 223 以说明向量执行单元 202 具有两个向量执行管线（如，双 SIMD 管线配置）的情况。本发明的实施例适于具有较大数目的向量执行管线（如，四个、八个、十六个等）的向量执行单元。

标量执行单元 201 为向量执行单元 202 提供数据和命令输入。在一个实施例中，标量执行单元 201 使用存储器映射命令 FIFO 225 来向向量执行单元 202 发送函数调用。向量执行单元 202 命令在此命令 FIFO 225 中排队。

命令 FIFO 225 的使用会有效地将标量执行单元 210 与向量执行单元 202 去耦。标量执行单元 201 可根据其自身的相应时钟运作，从而以其自身的相应时钟频率来操作，所述时钟频率可与向量执行单元 202 的时钟频率不同，且与向量执行单元 202 的时钟频率

分离地控制。

命令 FIFO 225 使向量执行单元 202 能够作为需求驱动单元来操作。例如，可将工作从标量执行单元 201 转交给命令 FIFO 225，且接着由向量执行单元 202 存取，从而以去耦异步方式进行处理。因此，向量执行单元 202 将按照标量执行单元 201 的需要或需求处理其工作负载。当不需要最大性能时，此功能将允许向量执行单元 202 保存功率（如，通过减少/停止一个或一个以上内部时钟）。

将视频处理功能划分成标量部分（如，由标量执行单元 201 执行）和向量部分（如，由向量执行单元 202 执行）允许将为视频处理器 111 建立的视频处理程序编译成单独的标量软件代码和向量软件代码。标量软件代码和向量软件代码可单独编译且随后链接到一起以形成连贯的应用程序。

所述划分允许向量软件代码函数被单独写入，且与标量软件代码函数不同。例如，向量函数可被单独写入（如，在不同时间，通过不同的工程师团队等）且可作为一个或一个以上子程序或库函数而提供，以供通过/利用标量函数（如，标量线程、处理等）使用。这允许标量软件代码和/或向量软件代码的单独独立更新。例如，向量子程序可与标量子程序相独立地更新（如，通过更新先前分配的程序，添加新特征以增加分配的程序的功能等），或反之亦然。通过标量处理器 210 与向量控制单元 220 及向量数据路径 221 的单独各自的高速缓冲存储器来促进所述划分，例如，标量处理器 210 的高速缓冲存储器为高速缓冲存储器 211-212，向量控制单元 220 及向量数据路径 221 的高速缓冲存储器为高速缓冲存储器 222-223。如上所述，标量执行单元 201 和向量执行单元 202 经由命令 FIFO 225 来通信。

图 3 展示根据本发明一个实施例的视频处理器 111 的示范性软件程序 300 的图。如图 3 所描绘，软件程序 300 说明视频处理器 111 的编程模型的属性，藉此由视频处理器 111 结合向量数据线程 302 来执行标量控制线程 301。

图 3 实施例的软件程序 300 的实例说明视频处理器 111 的编程模型，藉此标量执行单元 201 上的标量控制程序（如，标量控制线程 301）对向量执行单元 202 执行子程序调用（如，向量数据线程 302）。软件程序 300 的实例展示编译器或软件编程者将视频处理应用程序分解成标量部分（如，第一线程）和向量部分（如，第二线程）的情况。

如图 3 所示，在标量执行单元 201 上运行的标量控制线程 301 提前计算工作参数，并将这些参数馈入到向量执行单元 202，所述向量执行单元执行大多数处理工作。如上文所述，用于两个线程 301 和 302 的软件代码可单独写入和编译。

标量线程负责如下：

1. 与主机单元 204 介接并实施分类介接 (class interface)；
2. 向量执行单元 202 的初始化、设置和配置；及
3. 以回路形式在工作单元、程序块 (chunk) 或工作组中执行算法，使得通过每次迭代：
  - a. 计算用于当前工作组的参数；
  - b. 启动将输入数据传递到向量执行单元中的传递；和
  - c. 启动来自向量执行单元的输出数据的传递。

标量线程的典型执行模型是“即发即弃 (fire-and-forget)”。术语“即发即弃”是指这样一种属性：其中，针对视频基带处理应用程序的典型模型，将命令和数据从标量执行单元 201 发送到向量执行单元 202 (如，经由命令 FIFO 225) 且没有从向量执行单元 202 返回的数据，直到算法完成为止。

在图 3 的程序 300 的实例中，标量执行单元 201 将一直为向量执行单元 202 调度工作，直到命令 FIFO 225 中不存在任何空间 (如，!end\_of\_alg & !cmd\_fifo\_full) 为止。由标量执行单元 201 调度的工作计算参数并将这些参数发送到向量量子程序，且随后调用向量量子程序以执行所述工作。向量执行单元 202 对子程序 (如，vector\_funcB) 的执行在时间上被延迟，主要用于隐藏来自主存储器 (如，系统存储器 115) 的延迟。因此，视频处理器 111 的架构在向量执行单元 202 侧为指令和数据通信两者提供延迟补偿机构。将在下文更详细地描述这些延迟补偿机构。

应注意，软件程序 300 的实例在存在两个或两个以上向量执行管线 (如，图 2 的向量数据路径 221 和第二向量数据路径 231) 的情况下将会更复杂。同样地，软件程序 300 的实例在程序 300 被写入用于具有两个向量执行管线的计算机系统但仍保持可在具有单个向量执行管线上执行的能力的那些情形下将会更复杂。

因此，如上文图 2 和图 3 的论述中所描述，标量执行单元 201 负责在向量执行单元 202 上启动计算。在一个实施例中，从标量执行单元 201 传递到向量执行单元 202 的命令的主要类型如下：

1. 读取命令 (如，memRd)，其由标量执行单元 201 启动以将当前工作组数据从存储器传递到向量执行单元 202 的数据 RAM；
2. 从标量执行单元 201 传递到向量执行单元 202 的参数；
3. 执行命令，其呈待执行的向量量子程序的 PC (如，程序计数) 的形式；和

4. 写入命令（如，memWr），其由标量执行单元 201 启动以将向量计算的结果复制到存储器中。

在一个实施例中，一接收到这些命令，向量执行单元 202 就立即将 memRd 命令调度到存储器接口 203（如，以从帧缓冲器 205 读取所请求的数据）。向量执行单元 202 还检查所述执行命令并预取待执行的向量子程序（如果未存在于高速缓冲存储器 222 中）。

在此情形下向量执行单元 202 的目标是在向量执行单元 202 致力于当前的执行时，提前调度接下来的一些新的执行的指令和数据流。提前调度特征会有效地隐藏从指令/数据的存储器位置取指令/数据时涉及的延迟。为了提前进行这些读取请求，通过使用高速优化的硬件来实施向量执行单元 202、数据存储器（如，数据存储器 223）和指令高速缓冲存储器（如，高速缓冲存储器 222）。

如上文所述，数据存储器（如，数据存储器 223）充当向量执行单元 202 的工作 RAM。标量执行单元 201 感知数据存储器，并与其相互作用，仿佛它是 FIFO 的集合一样。FIFO 包括“流”，视频处理器 111 通过所述“流”进行操作。在一个实施例中，流通常是标量执行单元 201 启动传递（如，向向量执行单元 202）所进入的输入/输出 FIFO。如上所描述，标量执行单元 201 和向量执行单元 202 的操作被去耦。

一旦输入/输出流满了，向量控制单元 220 内的 DMA 引擎就停止处理命令 FIFO 225。这很快导致命令 FIFO 225 满。当命令 FIFO 225 满时，标量执行单元 201 停止向向量执行单元 202 发布额外工作。

在一个实施例中，向量执行单元 202 除了输入和输出流之外可能还需要中间流。因此，可将整个数据存储器 223 看作是关于与标量执行单元 201 进行相互作用的流的集合。

图 4 展示使用根据本发明一个实施例的视频处理器将视频与子图片混合的实例。图 4 展示视频表面与子图片进行混合且接着转换成 ARGB 表面的示范性情况。包括所述表面的数据作为亮度参数 412 和色度参数 413 驻留于帧缓冲存储器 205 中。子图片像素元件 414 也如所示驻留于帧缓冲存储器 205 中。向量子程序指令和参数 411 如所示例示于存储器 205 中。

在一个实施例中，每个流包括称作“瓷砖片”的数据的工作 2D 程序块的 FIFO。在此实施例中，向量执行单元 202 为每个流维持读取瓷砖片指针和写入瓷砖片指针。例如，对于输入流而言，当执行向量子程序时，向量子程序可从当前（读取）瓷砖片中进行消耗或读取。在后台中，数据通过 memRd 命令传递到当前（写入）瓷砖片。向量执行单元还可为输出流产生输出瓷砖片。接着，这些瓷砖片通过在执行命令之后的 memWr()命令

移动到存储器。这有效地预取瓷砖片且使其做好准备以对其进行操作，从而有效地隐藏延迟。

在图 4 的子图片混合实例中，向量数据路径 221 通过向量子程序指令和参数 411 的例示性情形（如，&v\_subp\_blend）来配置。这由线 421 展示。标量执行单元 201 在表面的程序块（如，瓷砖片）中读取，并使用 DMA 引擎 401（如，在存储器接口 203 内）将其加载到数据存储器 223 中。加载操作由线 422、线 423 和线 424 展示。

仍参看图 4，因为存在多个输入表面，所以需要维持多个输入流。每个流均具有相应的 FIFO。每个流可具有不同数目的瓷砖片。图 4 的实例展示以下情况：子图片表面处于系统存储器 115（如，子图片像素元件 414）中，且因此将具有额外的缓冲（如，n、n+1、n+2、n+3 等），而视频流（如，亮度 412、色度 413 等）可具有较小数目的瓷砖片。可根据流所经历的延迟的程度来调节缓冲器/FIFO 的数目。

如上文所述，存储器 223 利用先行预取（look ahead prefetch）法来隐藏延迟。由于此，随着针对适当的向量数据路径执行硬件预取数据，流可具有两个或两个以上瓷砖片的数据（例如，如 FIFO n、n+1、n+2 等所描绘）。

一旦加载数据存储器，就通过向量数据路径硬件 221 存取 FIFO，并通过向量子程序（如，子程序 430）对其进行操作。向量数据路径操作的结果包括输出流 403。这个输出流由标量执行单元 201 经由 DMA 引擎 401 复制回到帧缓冲存储器 205 中（如，ARGB\_OUT 415）。这由线 425 展示。

因此，本发明的实施例利用流处理的重要方面，所述重要方面是数据存储装置和存储器被抽象化为多个存储器瓷砖片。因此，流可被视为顺序存取的瓷砖片的集合。使用流来预取数据。此数据是瓷砖片的形式。预取所述瓷砖片以隐藏来自特定存储器源的延迟，所述数据源自所述特定存储器源（如，系统存储器、帧缓冲存储器等）。同样，可将所述流指定到不同位置（如，向量执行单元的高速缓冲存储器、标量执行单元的高速缓冲存储器、帧缓冲存储器、系统存储器等）。流的另一特征是其通常以先行预取模式存取瓷砖片。如上文所述，延迟越高，预取就越深且每流所用的缓冲就越多（例如，如图 4 所描绘）。

图 5 展示根据本发明一个实施例的向量执行单元的内部组件的图表。图 5 的图表展示从编程角度来看各种功能单元和向量执行单元 202 的寄存器/SRAM 资源的布置。

在图 5 的实施例中，向量执行单元 202 包括经优化以用于实施视频基带处理和执行各种编解码（压缩解压缩算法）的 VLIW 数字信号处理器。因此，向量执行单元 202 具

有许多针对于增加视频处理/编解码执行的效率的属性。

在图 5 的实施例中，所述属性包括：

1. 通过提供用于合并多个向量执行管线的选择的可调整的性能；
2. 每个管分配 2 个数据地址产生器 (DAG)；
3. 存储器/寄存器操作数；
4. 2D (x、y) 指针/迭代程序；
5. 较深管线 (如，11-12) 级；
6. 标量 (整数) /分支单元；
7. 可变指令宽度 (长/短指令)；
8. 用于操作数提取的数据对准程序 (aligner)；
9. 典型操作数和结果的 2D 数据路径 (4x4) 形状；和
10. 标量执行单元的从向量执行单元，其执行远程程序调用。

通常，编程者将向量执行单元 202 看作具有 2 个 DAG 503 的 SIMD 数据路径。指令以 VLIW 方式发布 (如，同时为向量数据路径 504 和地址产生器 503 发布指令) 且通过指令解码器 501 解码并分派到适当的执行单元。所述指令具有可变长度，其中多数普遍使用的指令以短形式编码。全指令集可具有长形式，如 VLIW 型指令。

图例 502 展示具有三个此种 VLIW 指令的时钟循回路。根据图例 510，最上的 VLIW 指令 502 包括两个地址指令 (如，用于 2 个 DSG 503) 和一个用于向量数据路径 504 的指令。中间 VLIW 指令包括一个整数指令 (如，用于整数单元 505)、一个地址指令和一个向量指令。最下的 VLIW 指令包括分支指令 (如，用于分支单元 506)、一个地址指令和一个向量指令。

向量执行单元可经配置以具有单个数据管或多个数据管。每个数据管由本地 RAM (如，数据存储器 511)、交叉矩阵 516、2 个 DAG 503 和 SIMD 执行单元 (如，向量数据路径 504) 组成。图 5 展示说明性目的的基本配置，其中仅例示 1 个数据管。当例示 2 个数据管时，其可作为独立线程或作为协作线程运行。

可经由地址寄存器文件单元 515 存取六个不同端口 (如，4 个读取和 2 个写入)。这些寄存器接收来自标量执行单元的参数或者整数单元 505 或地址单元 503 的结果。DAG 503 还充当集合控制器并管理寄存器的分配以处理数据存储器 511 的内容 (如，RA0、RA1、RA2、RA3、WA0 和 WA1)。交叉矩阵 516 经耦接以将输出数据端口 R0、R1、R2、R3 按任何次序/组合分配到向量数据路径 504 中，以实施给定指令。向量数据路径 504 的输

出可如所示被馈入回到数据存储器 511 中（如，W0）。常量 RAM 517 用于从整数单元 505 向向量数据路径 504 和数据存储器 511 提供经常使用的操作数。

图 6 展示描绘根据本发明一个实施例的存储器 600 的多个库 601-604 和具有瓷砖片 610 的对称阵列的数据存储器的布局。如图 6 中所描绘，为了说明性目的，仅展示数据存储器 610 的一部分。数据存储器 610 逻辑上包括瓷砖片的阵列。每个瓷砖片是 4x4 形状的子瓷砖片阵列。物理上，如存储器 600 所示，数据存储器 610 存储在存储器的“N”个物理库（如，库 601-604）的阵列中。

另外，数据存储器 610 视觉上描绘流中的逻辑瓷砖片。在图 6 的实施例中，这个瓷砖片是 16 个字节高且 16 个字节宽。这个瓷砖片是子瓷砖片的阵列（在此实例中为 4x4）。每个瓷砖片存储在物理库中。这在图 6 中在存在 8 个物理存储器库（如，库 0 到 7）的情况下，由每个 4x4 子瓷砖片内的数字所示。用使得在 2x2 的子瓷砖片配置中不存在普通库的方式在库中进行子瓷砖片的组织。这使得任何未对准的存取（如，在 X 和 Y 两个方向上）成为可能，而不会有任何库的冲突。

库 601-604 可经配置以支持对每个库的不同瓷砖片的存取。例如，在一种情况下，交叉矩阵 516 可从库 601 中存取瓷砖片的 2x4 组（如，库 601 的前两行）。在另一种情况下，交叉矩阵 516 可从两个相邻库中存取瓷砖片的 1x8 组。类似地，在另一情况下，交叉矩阵 516 可从两个相邻库中存取瓷砖片的 8x1 组。在每种情况下，DAG/控制器 503 可在交叉矩阵 516 存取所述库时接收所述瓷砖片，并以每时钟为基础向向量数据路径 504 的前端提供那些瓷砖片。

以此方式，本发明的实施例提供一种新的视频处理器架构，所述架构在有效地利用集成电路硅片面积、晶体管数、存储器速度要求等等的同时支持高级的视频处理功能。本发明的实施例维持高计算密度，并可容易地调整以处理多个视频流。本发明的实施例可提供许多高级的视频处理操作，例如（举例来说），MPEG-2/WMV9/H.264 编码辅助（如，回路内解码器）、MPEG-2/WMV9/H.264 解码（如，后熵解码（post entropy decoding））和回路内/回路外解块滤波器。

本发明的实施例提供的额外的视频处理操作包含（例如）高级的运动自适应去隔行、用于编码的输入噪声滤波、多相缩放/重新取样和子图片复合。本发明的视频处理器架构还可用于特定视频处理器-放大器（procamp）应用，例如（举例来说）色空间转换、色空间调节、像素点操作（如，锐化、直方图调整等）和各种视频表面格式转换。

概括而言且不作为限制，本发明揭示如下内容。描述了一种用于执行视频处理操作

的延迟容许系统。所述系统包含主机接口，其用于在视频处理器与主机 CPU 之间实施通信；标量执行单元，其耦接到主机接口且经配置以执行标量视频处理操作；和向量执行单元，其耦接到主机接口且经配置以执行向量视频处理操作。包含命令 FIFO 以使向量执行单元能够通过存取存储器命令 FIFO 而在需求驱动基础上操作。包含存储器接口以用于在视频处理器与帧缓冲存储器之间实施通信。在存储器接口中建立 DMA 引擎，以用于在多个不同的存储器位置之间实施 DMA 传递，并向命令 FIFO 中加载用于向量执行单元的数据和指令。描述了一种用于执行视频处理操作的视频处理器。视频处理器包含主机接口，以用于在视频处理器与主机 CPU 之间实施通信。包含存储器接口以用于在视频处理器与帧缓冲存储器之间实施通信。标量执行单元耦接到主机接口和存储器接口，并经配置以执行标量视频处理操作。向量执行单元耦接到主机接口和存储器接口并经配置以执行向量视频处理操作。描述了一种用于执行视频处理操作的用于视频处理器的多维数据路径处理系统。所述视频处理器包含经配置以执行标量视频处理操作的标量执行单元和经配置以执行向量视频处理操作的向量执行单元。包含数据存储存储器以用于存储向量执行单元的数据。数据存储存储器包含多个瓷砖片，所述瓷砖片具有以阵列形式布置的对称库数据结构。库数据结构经配置以支持对每个库的不同瓷砖片的存取。描述了一种用于执行视频操作的用于视频处理器的基于流的存储器存取系统。所述视频处理器包含经配置以执行标量视频处理操作的标量执行单元和经配置以执行向量视频处理操作的向量执行单元。包含帧缓冲存储器以用于存储标量执行单元和向量执行单元的数据。包含存储器接口，以在标量执行单元和向量执行单元与帧缓冲存储器之间建立通信。帧缓冲存储器包括多个瓷砖片。存储器接口实施瓷砖片的第一顺序存取并实施包括用于向量执行单元或标量执行单元的瓷砖片的第二顺序存取的第三流。

出于说明和描述的目的，已经提供对本发明的特定实施例的上文描述。上文描述不希望为排他的或将本发明限制于所揭示的精确形式，且根据上文的教示许多修改和变化都是可能的。选择并描述所述实施例以最好的解释本发明的原理和其实际应用，以藉此使所属领域的其它技术人员能够最好地用适于所预期的特定使用的各种修改来利用本发明和各种实施例。希望本发明的范围由所附的权利要求书和其等效物界定。

## 1 综述

VP2 是耦接到标量控制处理器的 VLIW SIMD 视频 DSP。其主要关注于视频编解码和视频基带处理。

### 1.1 VP2.0 的精髓

效率：VP2.0 在 perf/mm2 和 perf/mW 方面将是视频应用的计算效率较高的机器。

可编程性：其将会是高度可编程、易兼容且编程更安全的机器。

可调整性：VP2.0 的设计/架构应当可调整以匹配多个应用领域的性能要求。

## 1.2 设计目标

· 计算密度

○ 提供显著优于 VP1.0 的 perf/mm2 优点。

○ 有效实施新应用领域，如 H.264。

· HW 中的延迟容许解除了 SW 开发工程师的负担

○ 通过将存储器存取和计算重新排序来隐藏数据获取延迟。

○ 指令流的自动预取。

· 隐藏数据路径延迟

○ 中间结果的选择性转发。

○ 流计算模型。

· 可调整性：

○ 架构上 VP2 向量单元可将其数据路径向上调整 2x 且向下调整 1/2x。

○ 可通过选择性重设管线来实现频率改进。

## 1.3 应用对象

VP2.0 的设计和指令集将被优化从而非常有效地执行下列应用。

· mpeg2/wmv9/H.264 编码辅助（回路内解码器）

· mpeg2/wmv9/H.264 解码（后熵解码）

· 回路内/回路外解块滤波器。

· 高级的运动自适应去隔行

· 用于编码的输入噪声滤波

· 多相缩放/重新取样

· 子图片复合

· 处理放大（procamp）、色空间转换、调整、像素点操作（例如，锐化、直方图调整等）。

· 各种视频表面格式转换支持

架构上 VP2.0 可在下列领域有效

· 2D 图元、块图像传输（blit）、旋转等。

·基于精细化的软件运动估计算法。

·16/32 位 MAC 应用。

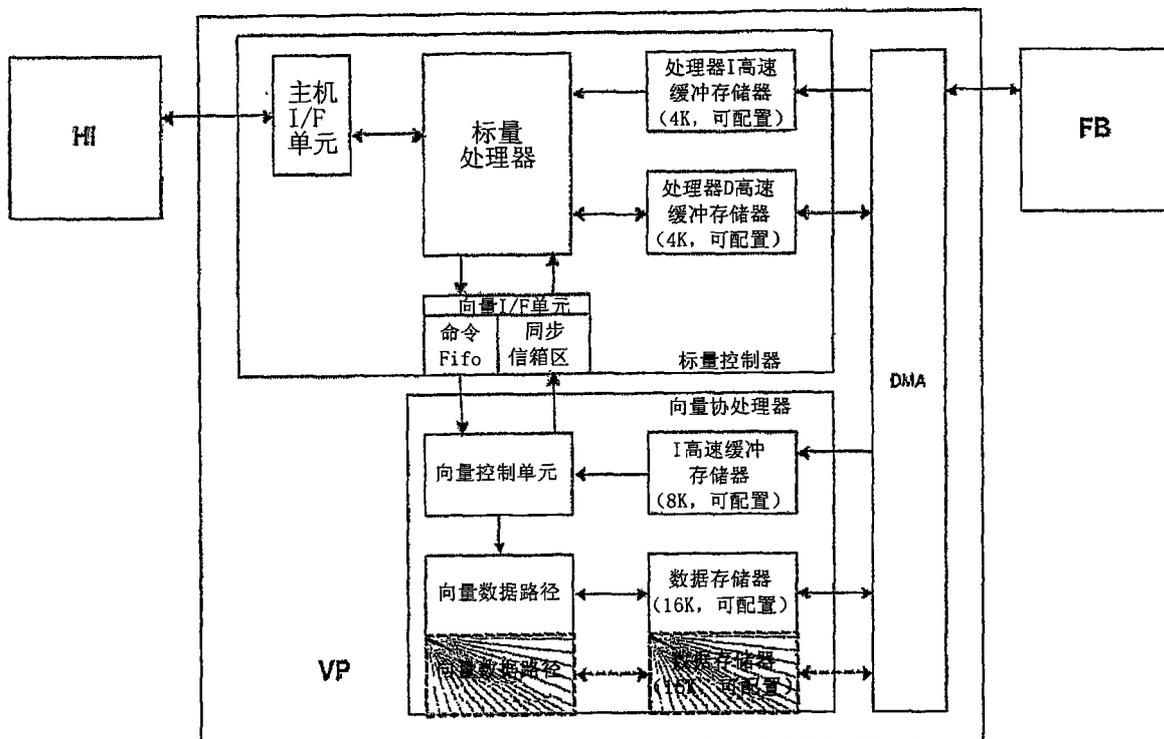
### 2 顶层架构

VP2.0 机器划分成标量和向量处理器。向量处理器充当标量处理器的从协处理器。标量处理器负责向向量处理器馈入控制流（参数、子程序自变量）并且还管理到达向量处理器中的数据 I/O。将在标量机器上执行算法的所有控制流，而将在向量处理器上执行实际的像素/数据处理操作。

标量处理器将是典型的 RISC 型标量器且向量协处理器是 SIMD 机器，其具有 1 或 2 个 SIMD 管（每个 SIMD 管具有 16 个像素数据路径）。因此，向量协处理器可产生高达 32 个结果像素作为原始计算能力。

标量处理器使用存储器映射命令 FIFO 向向量协处理器发送函数调用。协处理器命令在此 FIFO 中排队。标量处理器使用此 FIFO 与向量处理器完全去耦。标量处理器可根据其自身的时钟运行。向量处理器作为需求驱动单元操作。

VP2.0 的顶层图表给定如下。



VP2.0 程序可被编译成单独的标量代码和向量代码，且接着可随后链接在一起。分别地，向量函数可被单独地写入且可被提供为标量线程的子程序或库函数。标量处理器

具有其自身的指令和数据高速缓冲存储器。向量单元也具有指令高速缓冲存储器和数据 RAM（称为数据存储器）。这两个引擎被去耦，并通过 FIFO 进行通信。

### 3. 简单编程模型

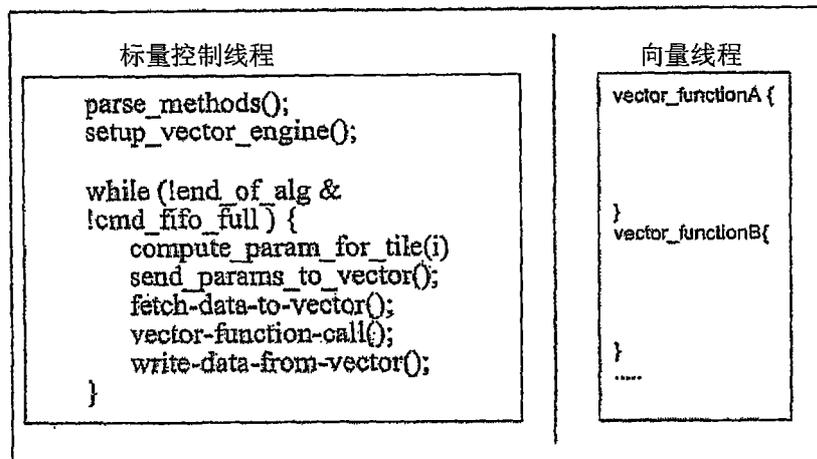
VP2.0 的最简单编程模型是对向量从协处理器执行子程序调用的标量控制程序。此处存在固有的假设，即编程者将所述问题分解成这 2 个线程。在标量处理器上运行的线程运行在提前计算工作参数并将其馈入到作为主要设备的向量处理器。期望这两个线程的程序被单独写入和编辑。

标量线程负责如下。

1. 与主机单元介接并实施分类介接。
2. 向量单元的初始化、设置和配置。
3. 以回路形式在工作单元、程序块或工作组中执行算法，使得通过每次迭代：
  - a. 计算用于当前工作组的参数；
  - b. 启动将输入数据传递到向量处理器中的传递；和
  - c. 启动来自向量处理器的输出数据的传递。

标量线程的典型执行模型是“即发即弃（fire-and-forget）”。期望此为视频基带处理的典型模型，其中没有从向量协处理器返回的数据。标量处理器将一直为向量处理器调度工作，直到在命令 FIFO 中存在空间为止。向量处理器对子程序的执行在时间上被延迟，主要是因为来自主存储器的延迟。因此，在向量侧提供延迟补偿机构很重要。在 VP2.0 中，向量处理器为指令和数据通信两者提供延迟补偿。在下文章节中概述所述机构。

典型的 VP 程序看起来如下



更复杂的编程模型是当我们具有 2 个数据管时。或当我们想要为 2 个数据管写入代

码，且使其在 1 个数据管机器上执行时。在章节 6 中研究所述编程模型。

#### 4. 流模型。

如同前面所概述的，标量引擎负责在向量处理器上启动计算。从标量引擎传递到向量引擎的命令的主要类型如下：

1. 读取命令 (memRd)，其由标量启动以将当前工作组数据从存储器传递到向量引擎的数据 RAM。

2. 从标量传递到向量的参数。

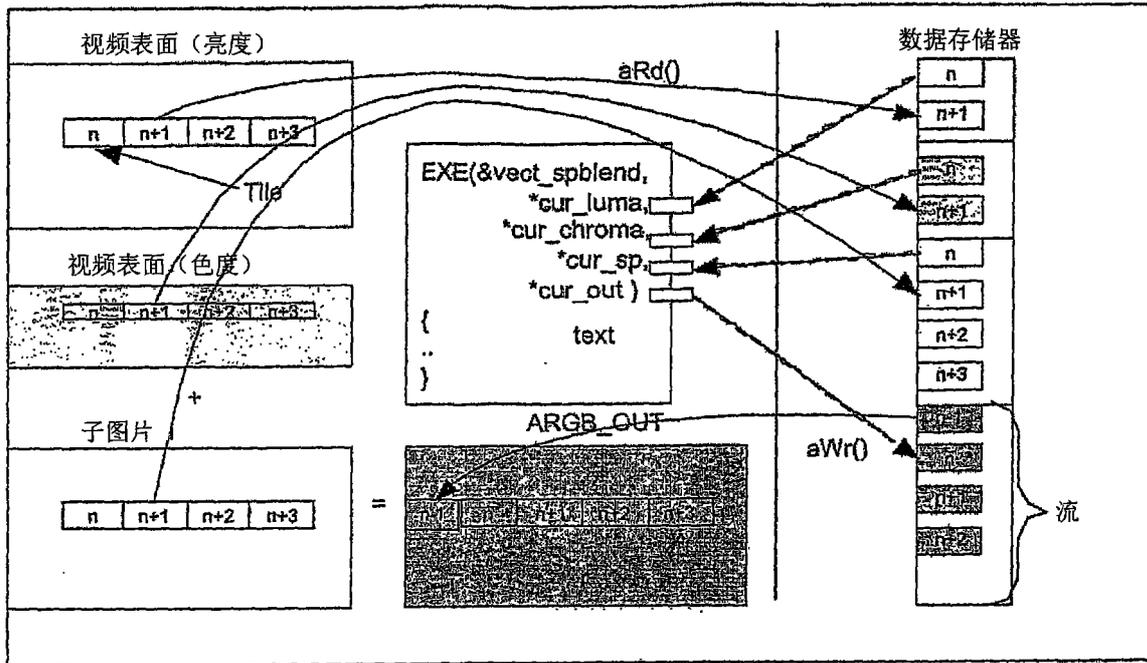
3. 执行命令，其呈待执行的向量子程序的 PC 的形式。

4. 写入命令 (memWr)，其由标量启动以将向量计算的结果复制到存储器中。

一接收到这些命令，向量处理器就立即将 memRd 命令调度到帧缓冲器 (FB) 接口。其还检查所述执行命令并预取待执行的向量子程序 (如果未存在于高速缓冲存储器中)。一个目标是当向量引擎致力于当前执行的同时，提前调度接下来的若干新的执行的指令和数据流。为了提前进行这些读取请求，向量引擎在硬件中管理数据存储器 and 指令高速缓冲存储器。

数据存储器是向量处理器的工作 RAM。标量处理器将此数据存储器看作是 FIFO 的集合或流。流基本上是标量启动传递所进入的输入/输出 FIFO。一旦输入/输出流满，那么向量 DMA 引擎就停止处理来自标量的命令 FIFO，很快使得其变满。因此，标量停止向向量引擎发布更多的工作。除了输入与输出流之外，向量可能还需要中间流。因此，整个数据存储器可看作是来自标量侧的流的集合。每个流是称作瓷砖片的工作的 2D 程序块的 FIFO。向量处理器为每个流维持读取瓷砖片指针和写入瓷砖片指针。对于输入流而言，当执行向量子程序时，向量子程序可从当前 (读取) 瓷砖片中进行消耗或读取。在后台中，数据通过 memRd 命令传递到当前 (写入) 瓷砖片。向量处理器还可为输出流产生输出瓷砖片。接着，这些瓷砖片通过在执行命令之后的 memWr() 命令移动到存储器。

通过将视频与子图片混合的实例来说明此模型。例如，考虑视频表面 (如，NV12 格式) 与子图片混合且接着被转换成 ARGB 表面的简单化的实例。这些表面驻留于存储器中。标量处理器读取这些表面的程序块 (瓷砖片) 并将其加载到数据存储器中。因为存在多个输入表面，所以我们不得不维持多个输入流。每个流可具有不同数目的瓷砖片 (如，在此实例中，我们可假定子图片表面处于系统存储器中，因此我们应对其进行更多缓冲)，而视频流可具有较少数目的瓷砖片。

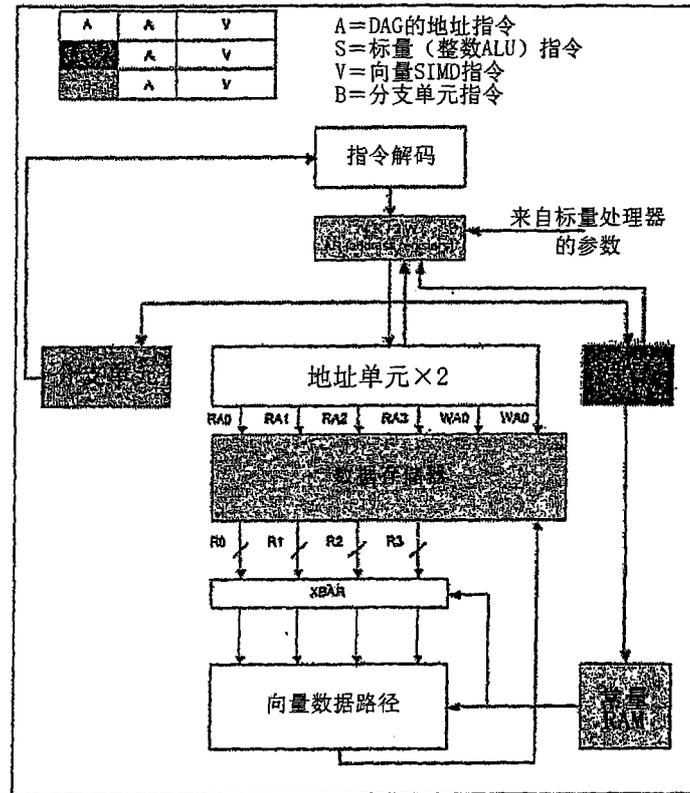


## 5. 向量协处理器

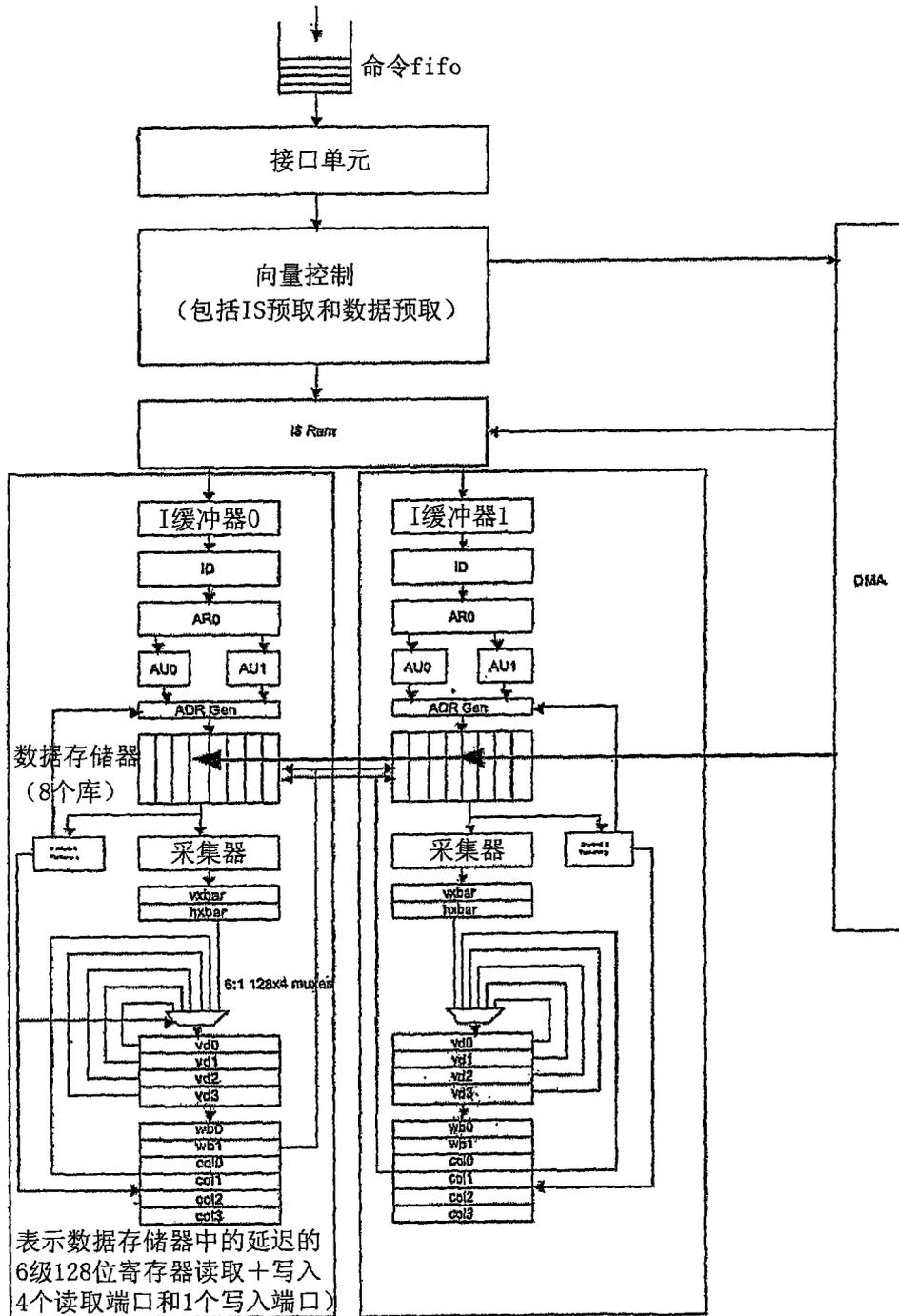
VP2 的向量协处理器是经设计而用于视频基带处理和编解码的 VLIM DSP。此处理器的一些重要设计属性包括：

1. 可调整性能，1 或 2 个数据管。
2. 每个管具有 2 个数据地址产生器 (DAG)
3. 存储器/寄存器操作数
4. 2D (x、y) 指针/迭代程序
5. 较深管线 (11-12) 级
6. 标量 (整数) /分支单元
7. 可变指令宽度 (长/短指令)
8. 用于操作数提取的数据对准程序 (aligner)
9. 典型操作数和结果的 2D 数据路径 (4×4) 形状
10. 标量处理器的从处理器，其执行远程程序调用。

在最简单的方面，编程者将向量协处理器看作是具有 2 个 DAG 的 SIMD 数据路径。指令以 VLIW 方式发布 (即，同时为向量数据路径和地址产生器发布指令)。所述指令具有可变长度，其中多数普遍使用的指令以短形式编码。全指令集可具有长形式。例如从编程者的角度来看，各种功能单元和寄存器/SRAM 资源的布置如下所示。



向量单元例示单数据管或双数据管。每个数据管由本地 RAM（数据存储器）、2 个 DAG 和 SIMD 执行单元组成。在基本配置中，仅存在 1 个数据管。当存在两个数据管时，其可作为独立的线程或作为协作的线程而运行。下文描述向量处理器的完整的管线图表。此处是具有 2 个数据管的完整配置。



### 6 高级编程模型

在章节 3 中，介绍 RPC 模型以说明所述基本架构。在此章节中，介绍了更高级的概念。

#### 6.1 双数据管配置

在双管配置中，共享处理器的下列资源。

- 标量控制器

- 向量协处理器中的向量控制单元
- 用于指令/数据获取的 DMA 引擎
- 指令高速缓冲存储器（可为双端口的）

复制下列资源

- 数据管（地址/分支/向量执行单元）
- 数据存储器
- 寄存器文件

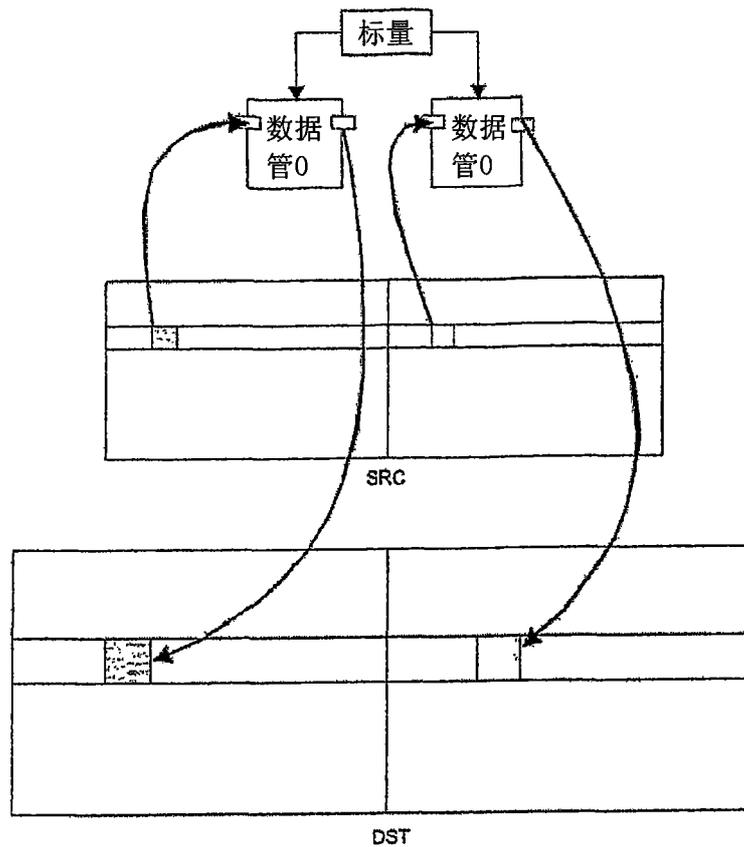
应注意：

1. 在仅有 1 个管的实例上，程序可被写入以用于 2 个管。向量控制单元将在同一物理管上映射每个管的执行。然而，因为两个管的流仅存在于一个数据存储器中，所以要调节数据存储器的大小。简单的方法是将流中的瓷砖片的瓷砖片大小或数目分两半。这将在配置时间通过标量线程来完成。存在如全局寄存器的复制和流映射这样的需要在微架构级解决的问题。

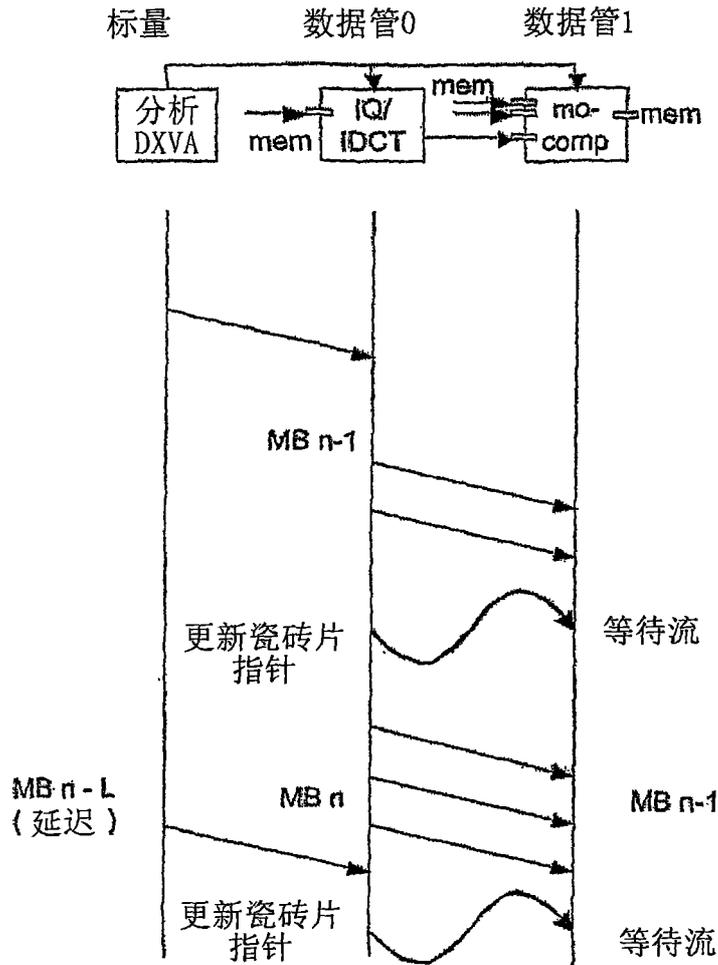
2. 写入以用于 1 个管的程序可在具有 2 个管的实例上运行。然而，这个代码将仅在一个管上运行且不使用另一个。机器将会半空闲。

3. 程序可被写入以用于 2 个管，每个管运行 2 个完全不同的线程。这不一定是优选的，因为我们仅具有非多线程的单个标量。因为我们仅支持一个标量执行线程，所以这可能不是优选的，然而，还是可以支持这个模型。

4. 程序可被写入以用于 2 个管，每个管运行相同线程。这是典型的希望用于可并行化算法（例如，大多数视频基带处理）的模型。这允许使用相同指令流对两个视频条或两个半个视频条等进行操作。每个数据管具有其自身的执行单元和数据存储器。标量控制器必须馈入 2 个数据管。然而，所述参数、读取和写入命令彼此相关（补偿（offset）），因此标量性能要求并不是正好两倍。这个模型的实例展示如下。



5. 可有在 2 个协作线程的情况下写入程序。这是期望用于编解码的模型，其中我们具有单个标量控制线程，但可能需要多个函数向量功能块连接在一起。这类似功能块的 direct-show 引脚模型 (pin model)。此应用的实例展示如下。因为我们仅具有 2 个数据管，所以这个模型仅被限制用于两个协作线程。另一警告是应在两个线程之间平衡线程。否则，存在性能损耗。在这些约束内，这个模型将在 2 个数据管上工作，且可调整回单个管。



6. 两个数据管可彼此同步。用于同步的基本方法是数据驱动。当数据可用于处理时，执行向量函数。通过从存储器进行读取或从其他数据管进行写入来填充所述流。一旦数据可用，向量控制单元将激活执行并运行它。所述流还可用作计数信号标（semaphore）。标量控制器和向量数据管均可递增并递减所述瓷砖片指针并使用流描述符作为计数信号标，甚至当不存在数据传递时也是如此。

补充综述：

通常，本发明的实施例执行如下：

1. 将媒体算法分解成标量和向量部分。

现货供应的标量设计，且其还向我们提供基于标量和向量部分的能力和性能要求而以不同时钟速度运行标量和向量部分的能力。

2. 流处理。
3. 2D 数据路径处理。
4. 延迟隐藏（针对数据和命令获取两者）

应用领域:

加密:

操作码隐藏

加密程序可仅位于芯片上。标量/控制器块仅请求执行特定操作，且加密引擎将获取所述指令等。因为标量甚至都不能看到什么算法在运行，所以其非常安全。其提供用于对用户隐藏加密算法的机构。

2D

VP2 指令集架构支持用于 2D 处理的指令。这些指令包括在许多 GUI/window 系统中使用的 ROP3 和 ROP4 支持。这允许媒体处理器在媒体处理器上运行 2D 操作。此处的固有优点是节省功率。

ISA

作为指令槽的条件代码:

针对条件代码操作，我们具有单独的发布槽 (issue slot) (在我们的多个发布指令束中)。现有技术是人们使用 SIMD 指令，所述 SIMD 指令也可影响条件代码/断言寄存器。但是通过在 VP2 中所采用的方法，数据处理和断言寄存器处理可独立地进行调度，所以会导致较高性能。

存储器 I/O

微码 DMA 引擎:

DMA 引擎可经编程 (或可具有自身较小的微码) 来执行各种操作，如，用于流的数据预取、对数据格式进行格式化、边缘填充等。通常，可编程 DMA 引擎且非接线功能。因此存储器 I/O 处理器与媒体处理核心的结合增加了总体的系统层性能。媒体处理器核心不再必须进行数据 I/O 处理。

存储器阶层架构:

在 VP2 架构中，优化存储器阶层以最小化存储器 BW 并提供延迟补偿。提供许多不同方案，例如

-流数据存储器的第一层，其作为擦除随机存取存储器对于向量核心来说是可见的。由 HW 管理以先行进入由标量处理器产生的请求流。这个存储器视情况可由 L2 高速缓冲存储器支持，以用于数据再利用。L2 高速缓冲存储器可以流为基础划分成各区段。

-L1 高速缓冲存储器，其由流存储器支持。数据存储器预取下一个相关数据组。

-使用流指针作为数据标记的高速缓冲存储器。

-使用标量产生的流地址来预取/高速缓存 L1 数据存储器和 L2 高速缓冲存储器。

优化的标量-向量通信链路:

MemRd/Wr 格式:

压缩来自标量的命令以用于将系统存储器读取并写入到本地存储器中。节省管理 DMA 引擎需要的控制流带宽。同时, 不限制所支持的事务的类型。

针对向量 L2 推测标量 2 向量:

参数压缩支持减少通信带宽的参数修改程序和迭代程序。

管线高速缓冲存储器:

管线指令高速缓冲存储器。支持多种方案, 例如

-通过追踪在向量与标量处理器之间运行的执行来管理每个高速缓冲存储器线的生存周期。这允许指令在向量处理器开始执行之前准备就绪。如果指令不是已经存在于高速缓冲存储器中, 那么对其进行预取。

-对于较小延迟配置而言, 通过使指令变成小 FIFO 来将指令高速缓冲存储器最小化。已经存在于 FIFO 中的执行可被再利用, 否则, 其可被再次获取。

整体架构:

可在各种处理元件之间共享数据存储器。这些通过流进行通信, 且可彼此馈入。这一架构可具有一组异类的功能单元, 如 SIMD 向量核心、DMA 引擎、通过流连接的固定功能单元。

计算/DP

任意/灵活的形状/半个管:

数据路径以可变形状操作。数据路径的形状可经配置以匹配问题组 (problem set)。通常人们使用 1D 数据路径。VP2 可处理可为可变大小 4×4、8×4、16×1 等的形状来匹配算法。

可调整性:

VP2 数据路径架构使用指令护送技术 (注意, 我们具有 16 路 SIMD 管, 其中每个操作数为 1 个字节宽。我们可具有 8 路 SIMD 管 (使 2 个管为一组) 且具有较宽的 SIMD 数据路径, 其中每个操作数为 2 个字节, 且同样地, 我们可具有 4 路 SIMD 管 (使 4 个管为一组) 且具有较宽的 SIMD 数据路径, 其中每个操作数为 4 个字节。) 以在较窄的数据路径上通过多个循回路来执行较宽的 SIMD 指令, 从而节省面积。

如, VP2 可将数据路径从 16 路 SIMD 调整为 8 路 SIMD。

耦接字节通道。

耦接 SIMD 路以增加操作数宽度。例如，当前 16 路 SIMD 具有 8 位操作数。对于 8 路 SIMD，可将其增加到 16 位操作数，且对于 4 路 SIMD，可将其增加到 32 位操作数。

#### SIMD 地址产生器

用于 SIMD 管的每个路的单独的流地址产生器。

VP2 可使用 SIMD 地址产生器，它的请求可结合到对数据存储器的最小存取中。

#### 使用交叉矩阵和采集器的数据扩展

有能力使用交叉矩阵来产生更多的数据操作数。减少读取端口压力并节省功率。

#### X2 指令：

不是所有的指令都可在数据路径中使用所有的 HW 元件（加法器/乘法器）。所以对于简单的指令（如，add/sub），我们可比复杂的指令处理更宽的数据形状。所以代替将性能限制到最小共同大小，VP2 使用灵活的指令集，只要读取端口可维持所述操作带宽，其就寻求机会来试图以较宽形状操作。

#### 多线程/多核心媒体处理。

VP2 架构支持各种多线程选项，例如

- 多线程标量处理器在通过流连接的多个向量单元上调度程序调用。
- 多个线程根据指令/通过指令或者通过执行线程切换在单个向量引擎上运行。

#### 使用不同向量/标量进行功率管理

通过去耦的标量和向量部分，你可基于功率和性能要求以不同速度运行这 2 个块。

#### 上下文切换：

这个媒体处理器由于其不具有寄存器的架构而具有支持非常快的上下文切换的能力。存在 HW 支持，用以追踪标量 2 向量命令队列和将其保存并重放以实现上下文切换。同样可在页面故障时启动上下文切换。

这使得媒体处理器能够维持实时处理任务，如输入/输出显示处理，同时能够支持非实时任务（如 2D 加速或准时视频增强）以馈入显示管线。

这个上下文切换能力连同其指令集允许 VP2 为统一的像素/编解码处理。

#### 数据存储器组织：

VP2 使用具有下列管线的数据存储器组织

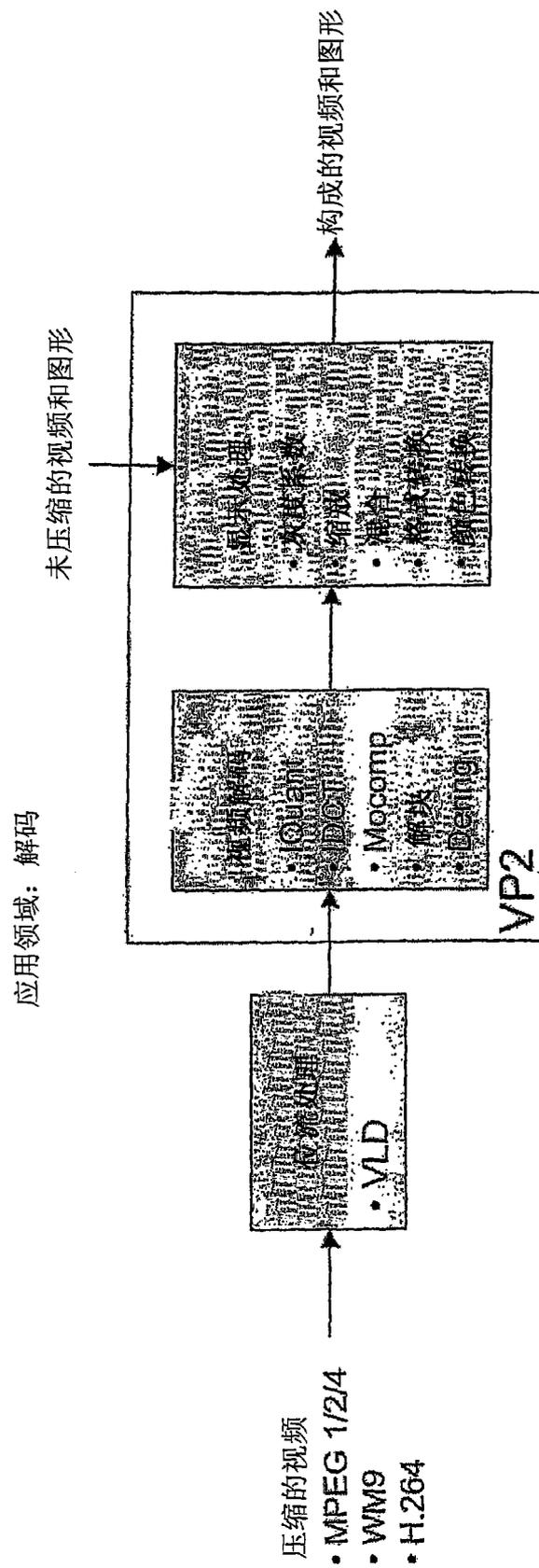
每个方向上有高达 16 个像素可被存取，而不存在库冲突。完成上述动作的同时可保持跨距请求最小。

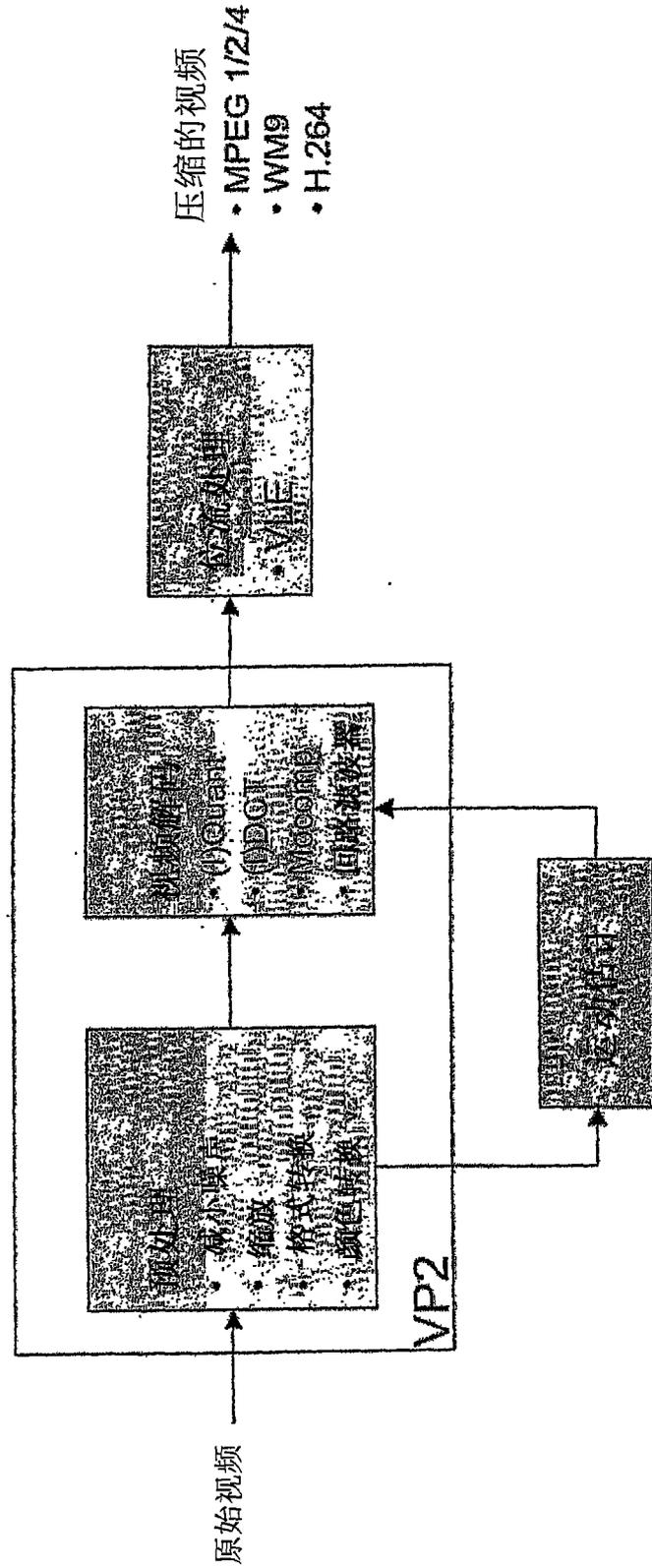
数据存储器组织允许数据形状的有效调换。

2D 寻址是所支持的内部数据存储器，在大多数媒体处理应用（如，视频）中消除了线性地址的 SW 计算。



应用





架构

## 高密度计算

- VP1架构的 $2\times$  perf/mm<sup>2</sup>
- 在新的应用上有效（与固定功能HW相比）

## 可编程

- “适当”编译：针对标量的C，针对向量的内联函数
- 在HW中管理延迟隐藏
- 保护挂起和坏地址

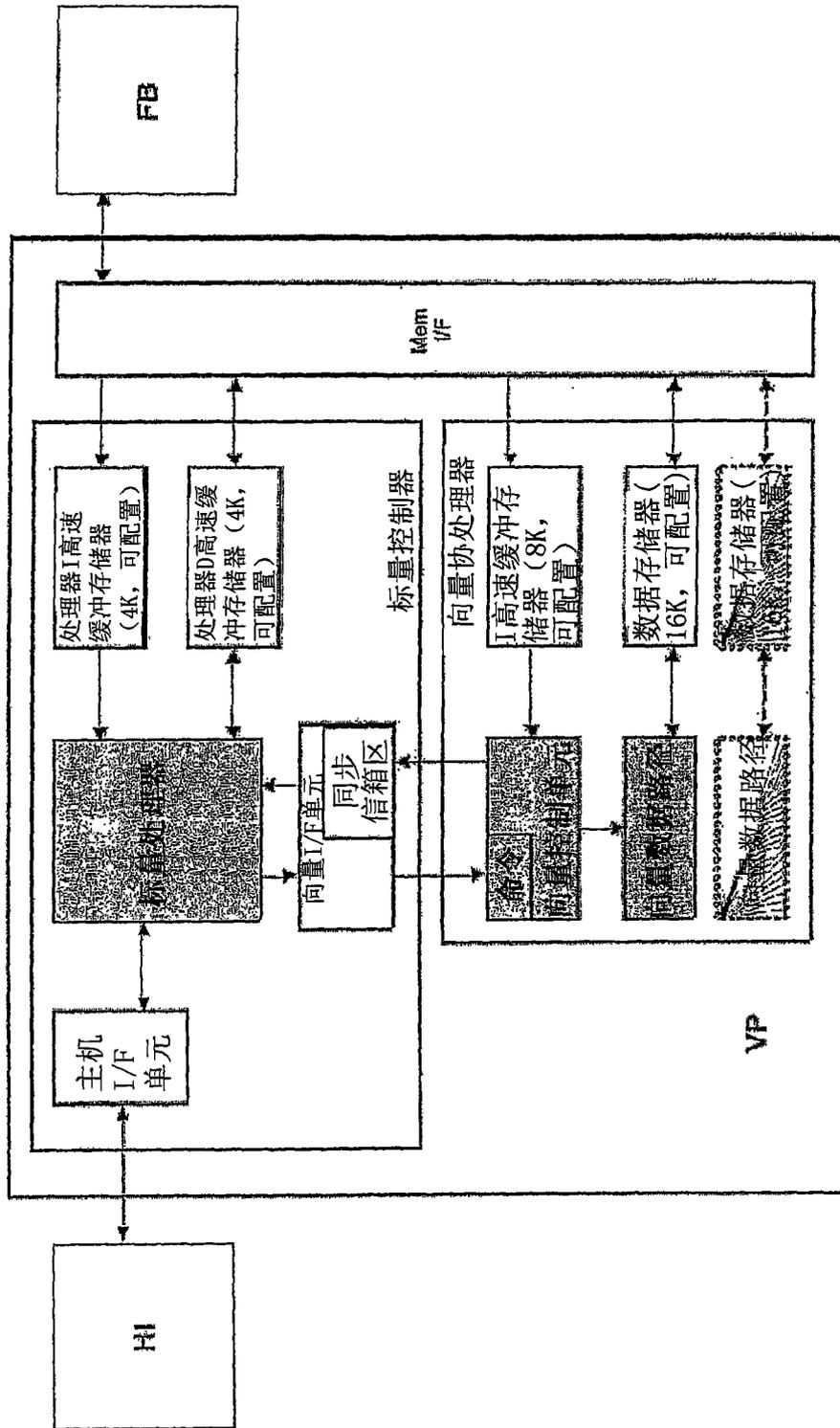
## 可缩放

- $2\times$ 和 $_{\times}$ 缩放选项
- 可经由时钟频率缩放

## 高精度操作支持

- 10b+20b 整数数据类型

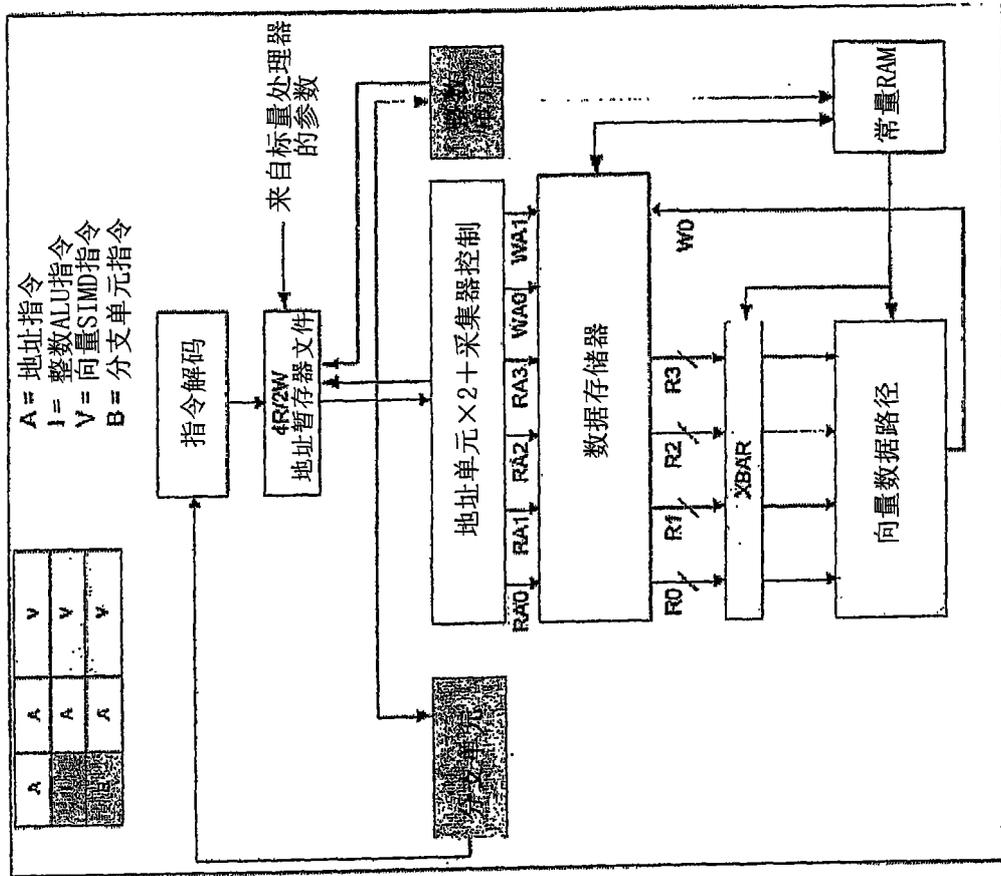
顶层块图表



## 块综述

- 标量处理器:
  - 程序流量控制
  - 瓷砖片光栅化和地址计算
  - 松散耦接向量单元的馈入
- 向量处理器:
  - 3发布VLIW (V+1、A、B)
  - 4×4 SIMD数据路径
  - 直接来自数据存储器向量的向量操作数 (无暂存器文件传递)
- 标量/向量接口:
  - 松散耦接的处理器-标量馈入命令fifo
  - 由HW管理的指令和数据流 (延迟隐藏)
  - 当指令和数据准备就绪时发布向量命令

向量数据路径



## 从VP1继承的

- 点乘积定向的数据路径
- VP2从2个分接头到4个分接头数学
- 具有A/I/B/V指令的VLIW。（修改的发布规则）
- 数据路径架构：累积器、数据转发、HW互锁
- 自由调换。从 $16 \times 16$ 变化到 $>4 \times 4$
- 专用地址单元
- 复杂指令集（针对视频进行优化）

VP2特征1

- 去耦的标量处理器
- C中可编译的外部回路代码的良好对象
- 指令流
- HW管理的指令预取 (使用管线I\$)
- 应用可比指令高速缓冲存储器具有更大的覆盖区
- 数据预取
- 在数据存储器中的HW管理的流
- 编程者不必须作外部回路展开
- 精度改进
- 10b单精度
- 新的20b双精度
- 更好的ISA
- 更好的误差管理
- 存储器保护, 指令截留

## VP2特征2

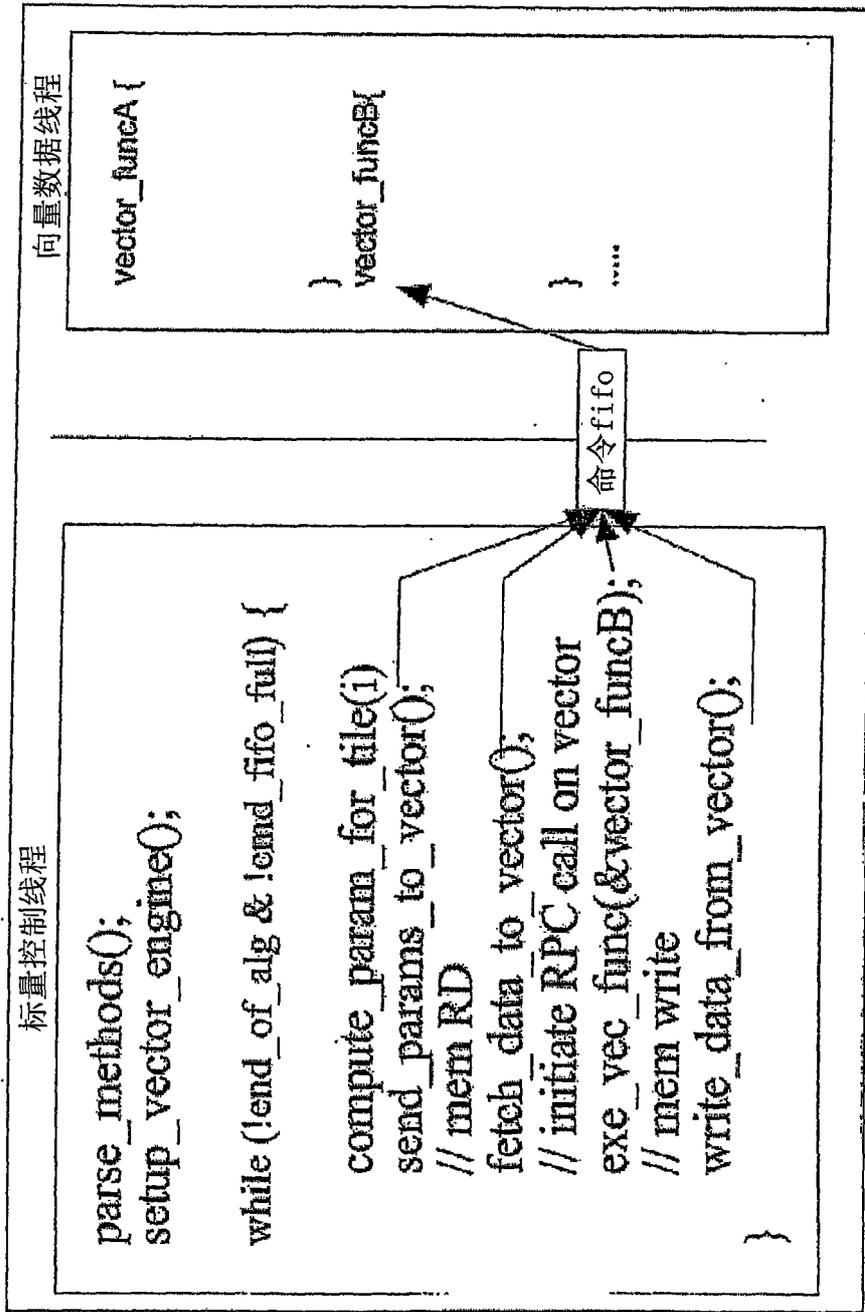
-  4×4 SIMD数据路径
  -  VP1是16×1 SIMD机器
  -  对高级编解码 (H.264/WMV9) 的非常有效的组织
  -  降低2D处理的带宽要求
-  改进的数据存储器组织
  -  用来匹配数据路径并比VP1提供远为廉价的调换能力的4×4结构
-  用来仿真多端口RAM的采集器结构 (如SPA)
  -  向量管直接从数据存储器进行操作 (无向量暂存器文件)
-  专用的交叉矩阵级
  -  提取未对准的操作数
  -  从若干读取端口创建多个操作数
-  常量RAM
  -  减小数据存储器读取带宽压力
-  非常灵活的条件代码/断言支持

目标应用

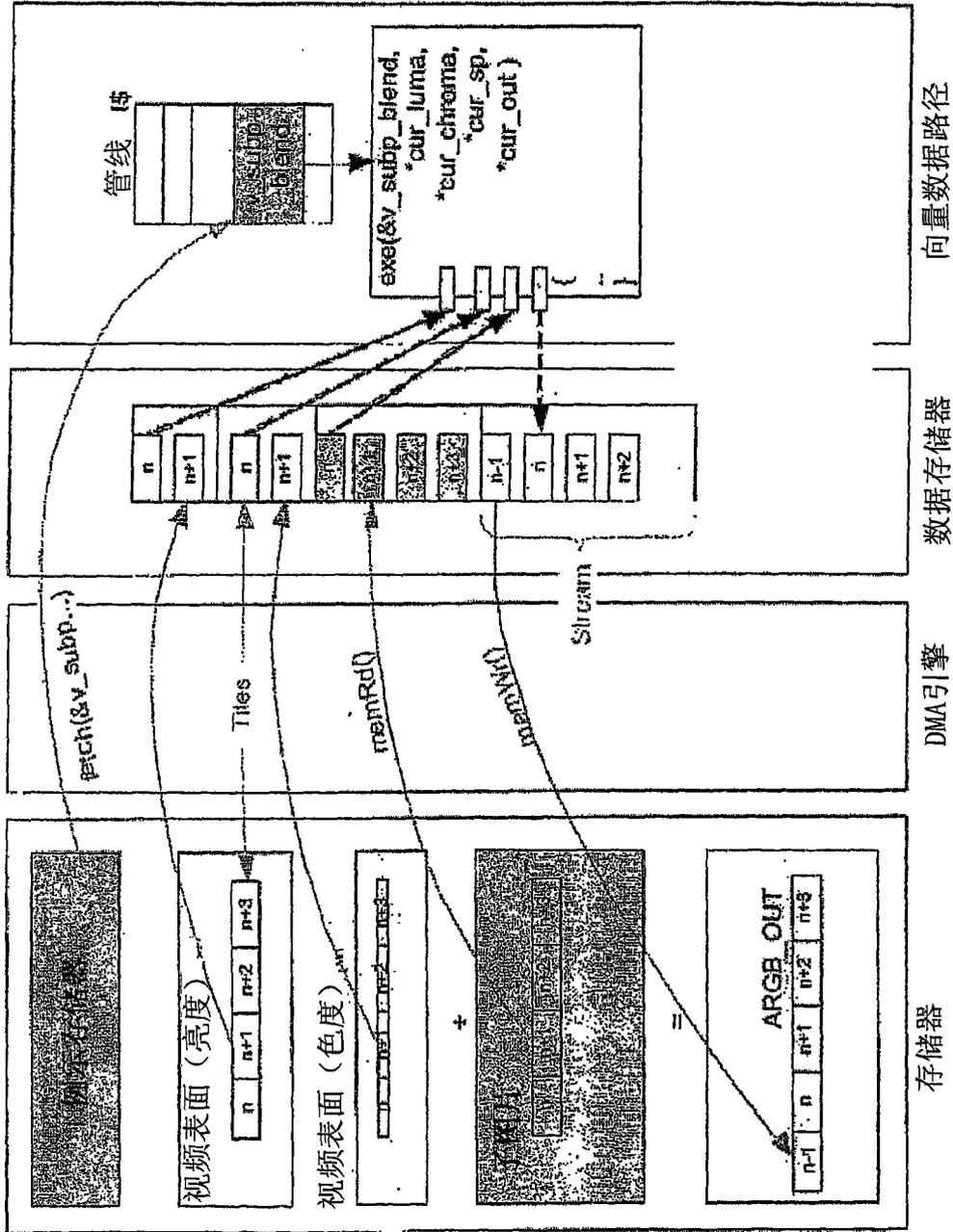
-  编解码
  -  Mpeg2/wmv9/H.264编码辅助 (回路内解码器)
  -  Mpeg2/wmv9/H.264解码 (后VLD解码)
  -  回路内/回路外解块滤波器
  
-  图像处理/增强
  -  高级的运动自适应去隔行
  -  用于编码的输入噪声滤波
  -  多相缩放/重新取样
  -  子图片复合
  -  按像素变换: 处理放大、色空间转换、灰度系数、LCD过驱动、直方图调整等
  
-  视频表面格式转换

编程模型

编程模型实例



流和延迟隐藏



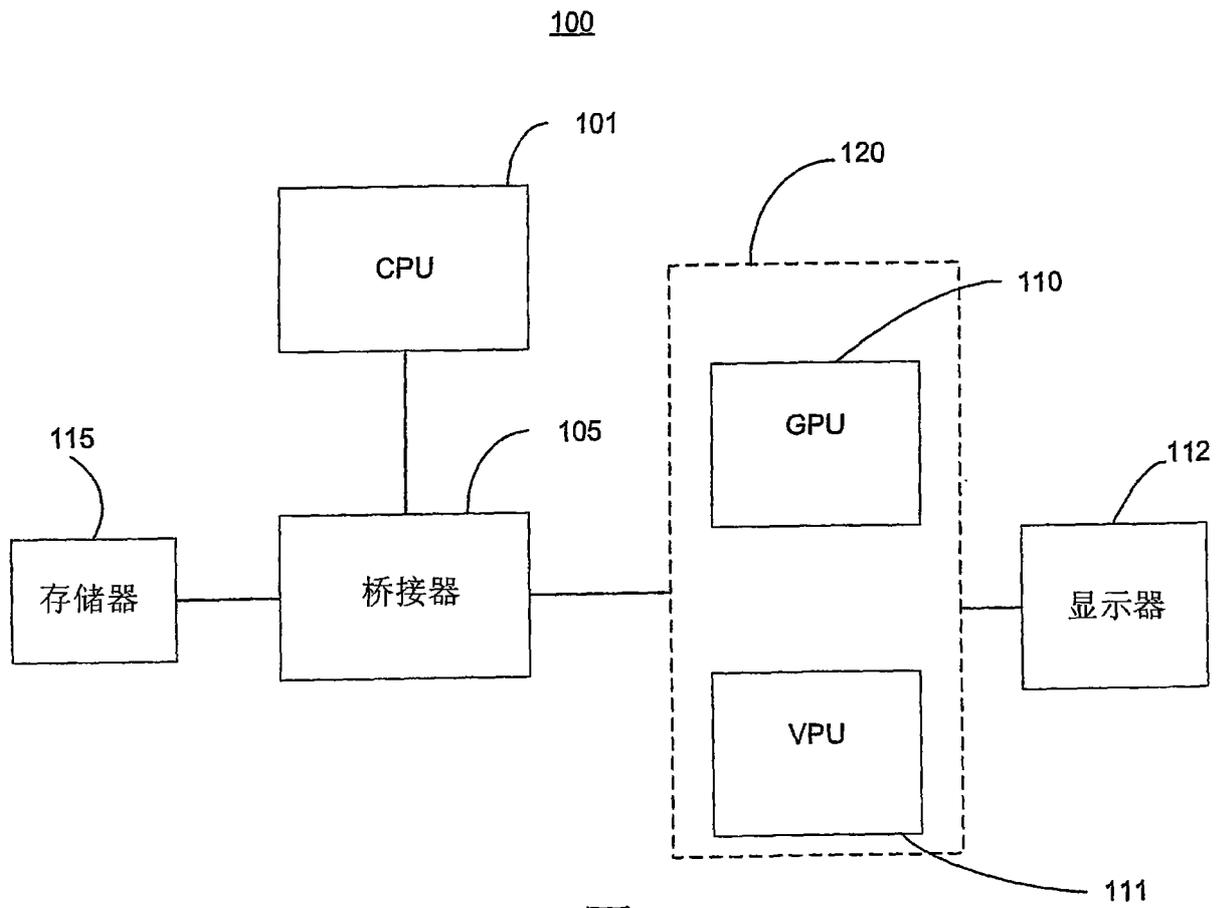


图1

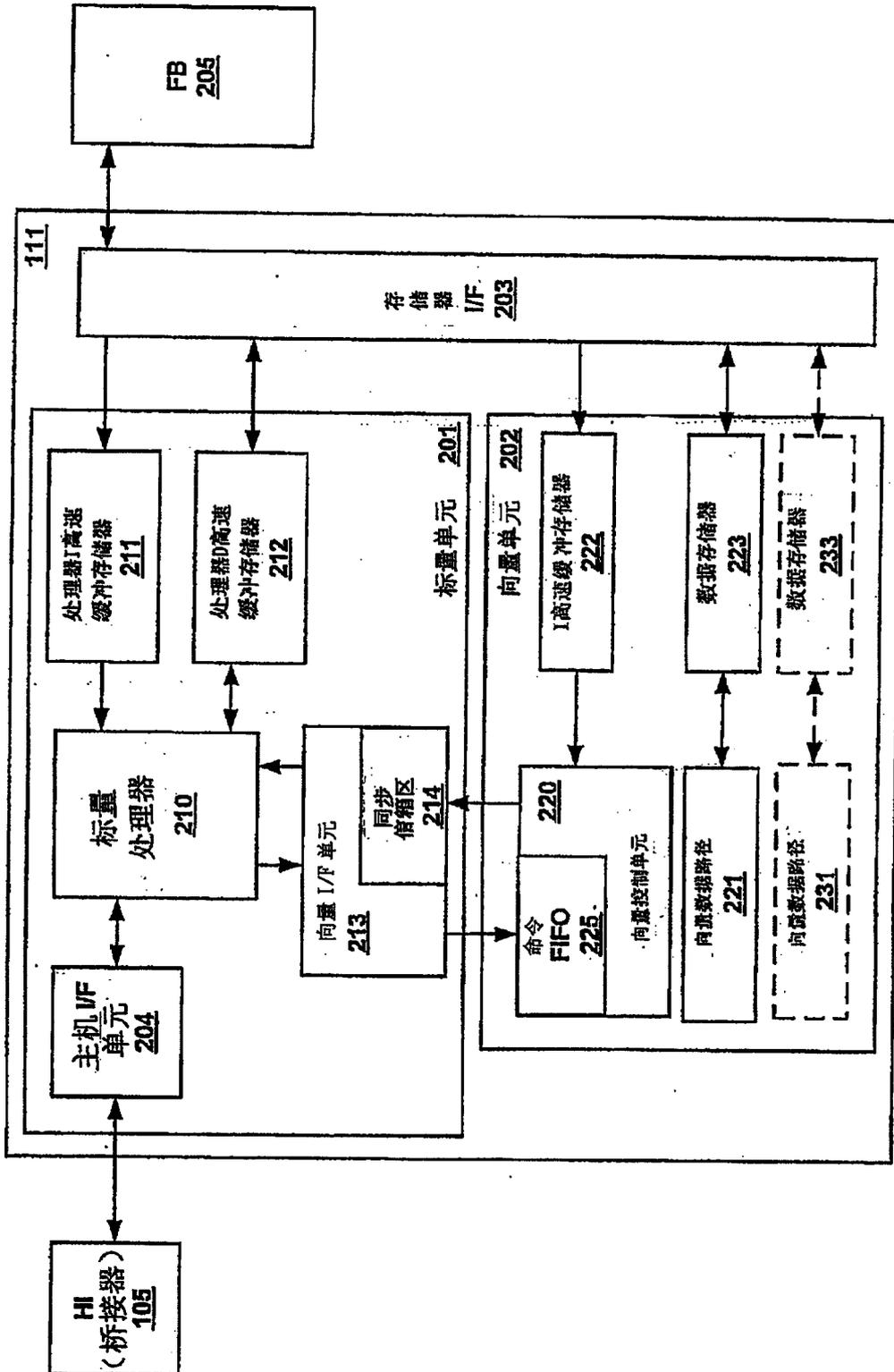


图2

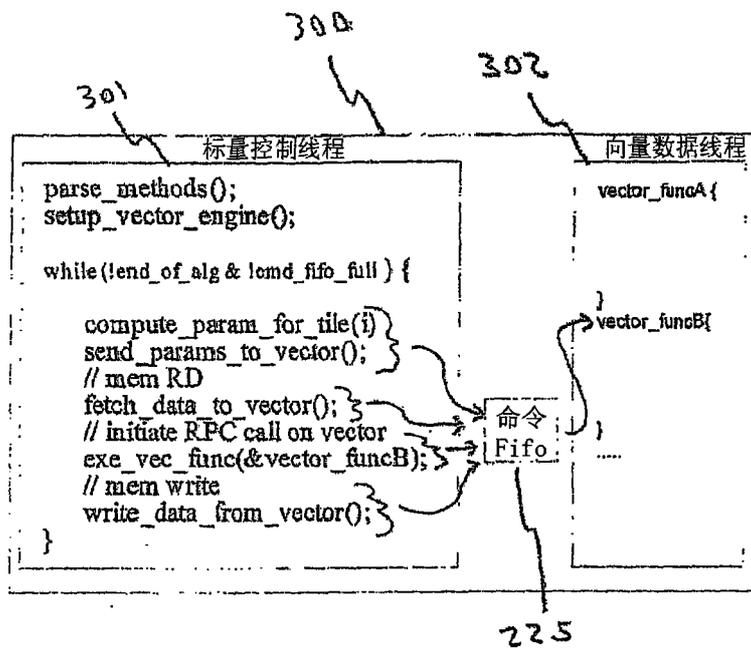


图3

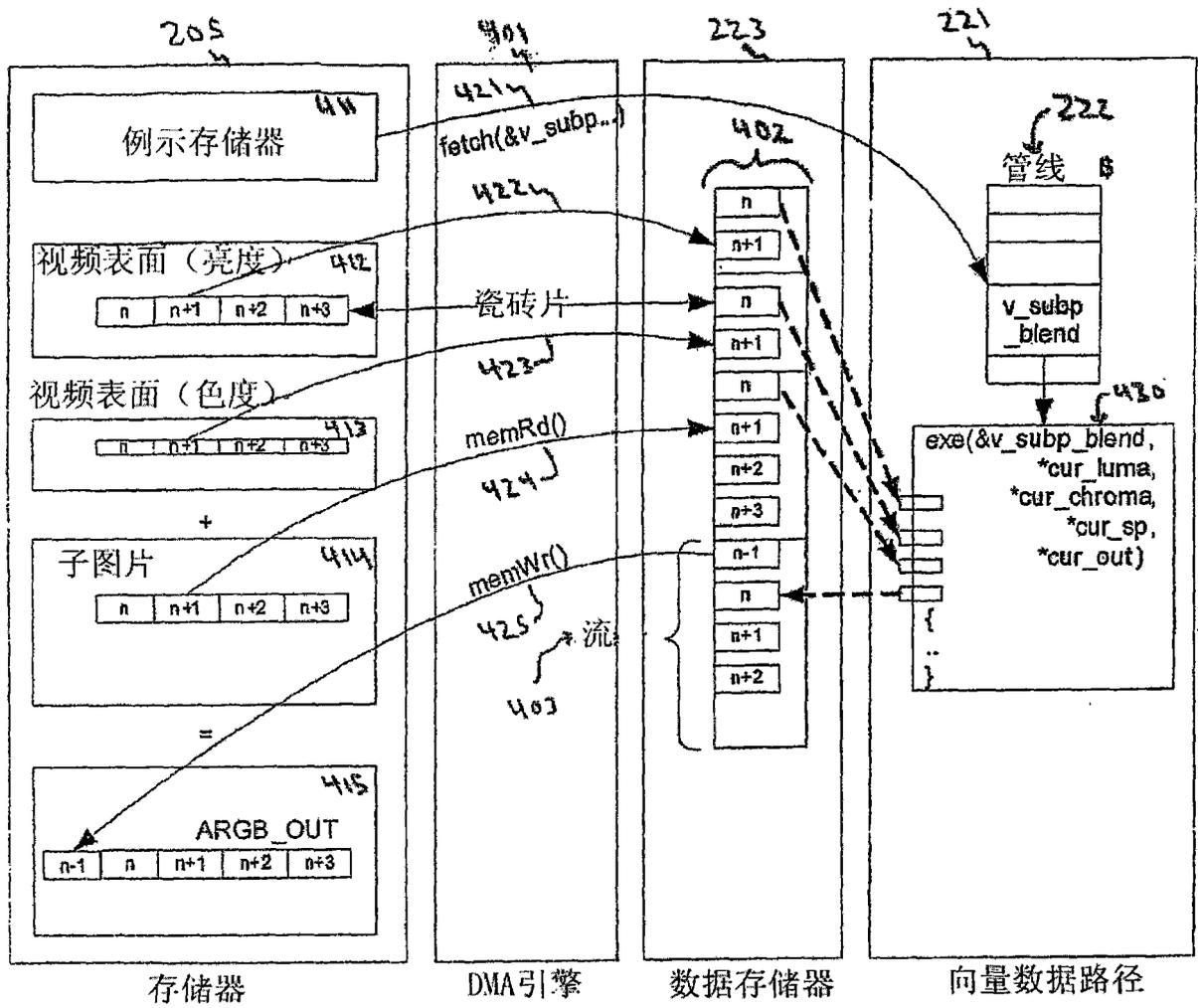


图4

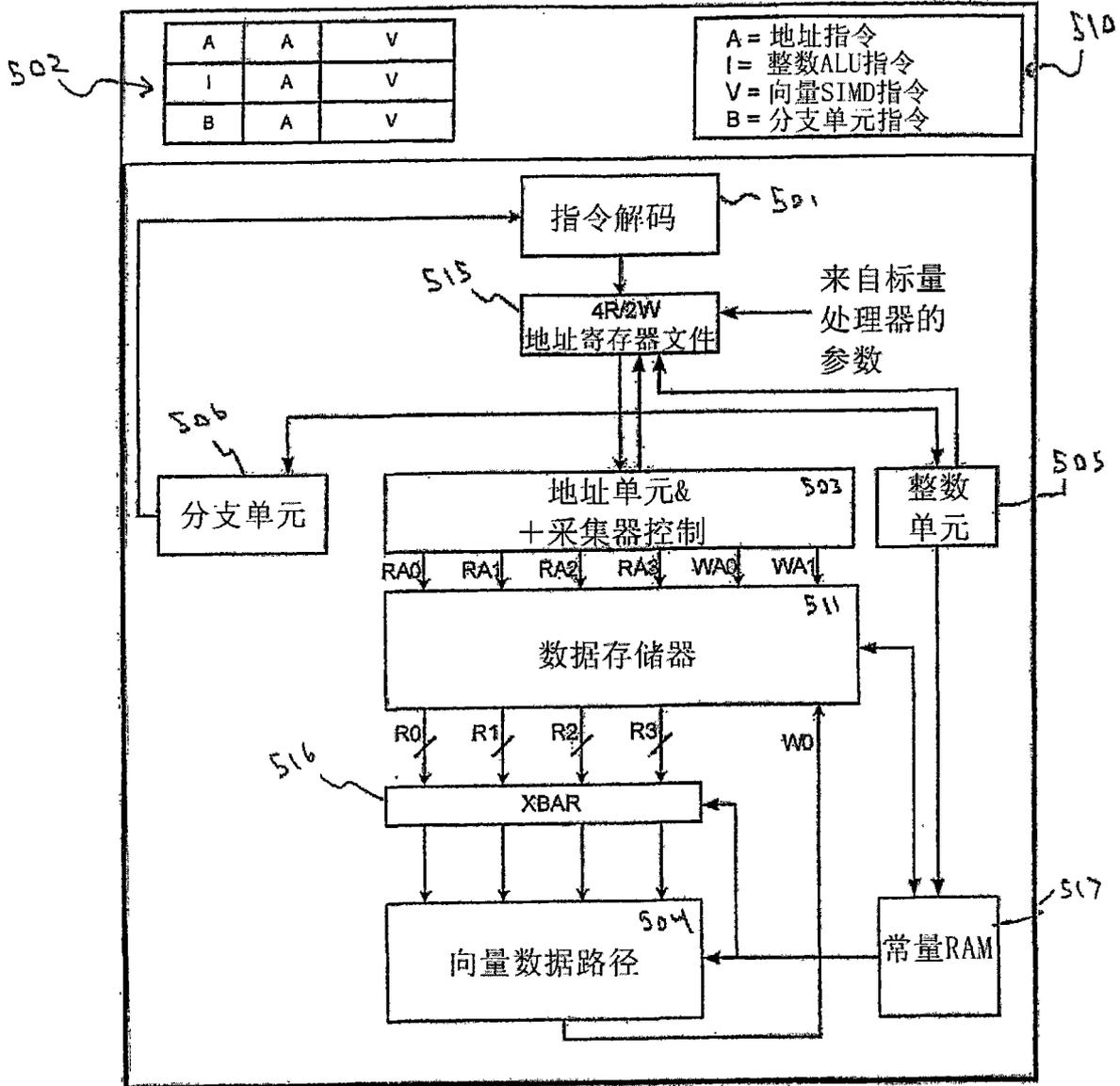


图5

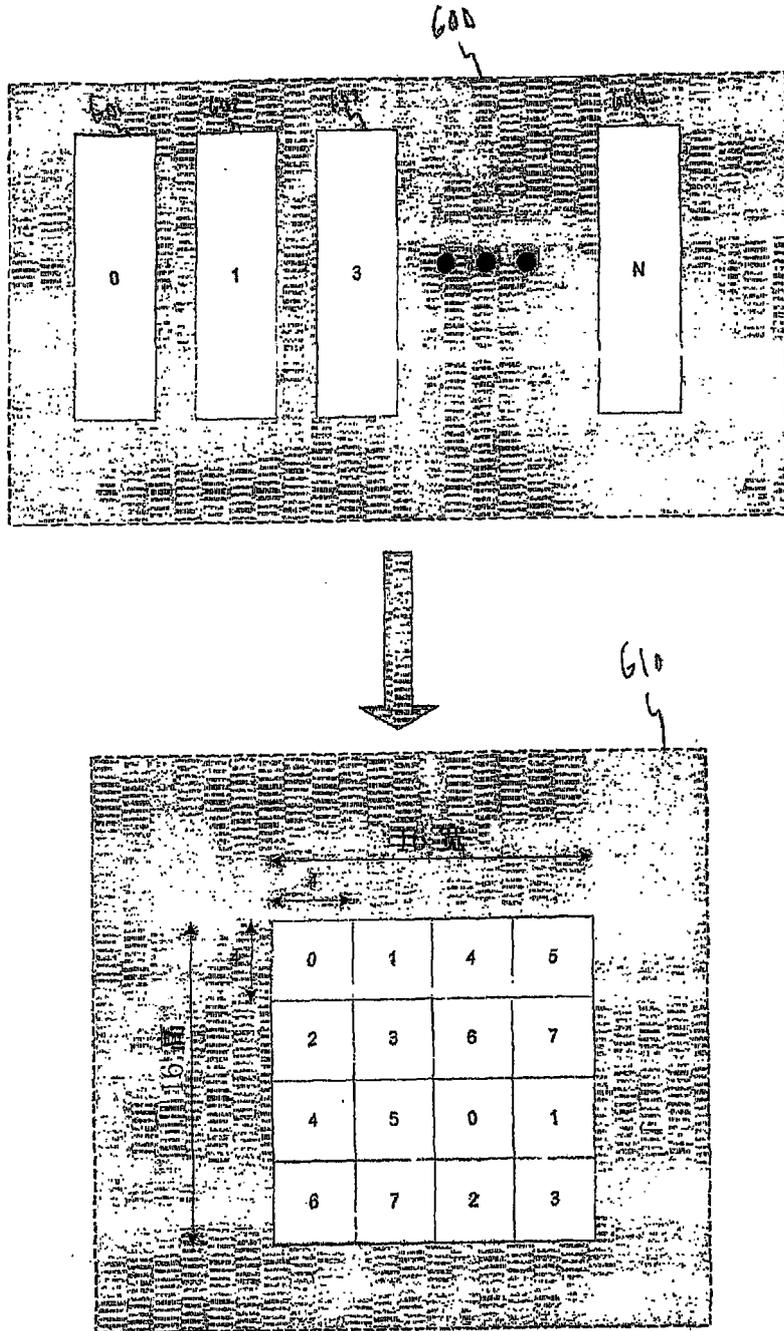


图6