

[19] 中华人民共和国国家知识产权局

[51] Int. Cl<sup>7</sup>

G06F 3/00

G06F 9/45



# [12] 发明专利说明书

[21] ZL 专利号 02114546.6

[45] 授权公告日 2004 年 9 月 15 日

[11] 授权公告号 CN 1166995C

[22] 申请日 2002.4.27 [21] 申请号 02114546.6

[71] 专利权人 西安交通大学

地址 710049 陕西省西安市咸宁路 28 号

[72] 发明人 郑南宁 吴 勇

审查员 杨 蕊

[74] 专利代理机构 西安通大专利代理有限责任公  
司

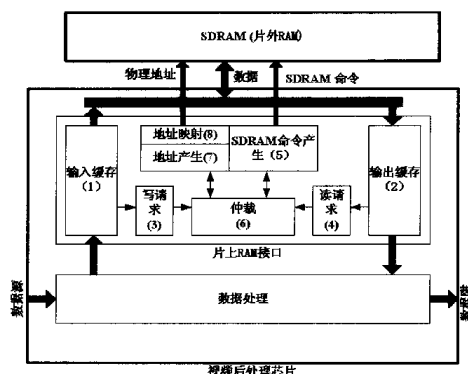
代理人 李郑建

权利要求书 2 页 说明书 9 页 附图 1 页

[54] 发明名称 高速视频处理接口控制器及其处理方法的降低。

## [57] 摘要

本发明公开了一种高速视频处理接口控制器及其设计方法，由输入缓存 (Input Buffer)、输出缓存 (Output Buffer)，读请求 (Read Beg)，写请求 (WriteBeg)，SDRAM 命令产生 (SDRAM Command Gnerator)，仲裁 (Arbitrate)，地址产生 (Address Generator) 及地址映射 (Address Map) 几部分功能模块组成。片外帧存储器存贮了视频处理所需要的几帧图像，接口中的输入、输出缓存在芯片内的处理部分与芯片外的 SDRAM 帧存储器之间形成数据交换接口，芯片外 SDRAM 的读写操作的发生是通过“被动呼唤”的设计策略完成的。在运动估计的算法实现中：(1) 只与每一执行的时间循环同步，即同步于每一新帧的开始；(2) 数据输入和输出的次序一定且不变；(3) 在同一时间内需要得到 2 帧的视频数据；采用本发明能够使帧存储器存取功



1. 一种高速视频处理接口控制器，其特征在于，高速视频处理接口控制器由输入缓存模块[1]、输出缓存模块[2]，写请求模块[3]，读请求模块[4]，SDRAM命令产生模块[5]，仲裁模块[6]，地址产生模块[7]及地址映射模块[8]组成；输入缓存模块[1]与写请求模块[3]连通，输出缓存模块[2]与读请求模块[4]连通，读、写请求模块与仲裁模块[6]相连，仲裁模块[6]分别与相互连接的SDRAM命令产生模块[5]、地址产生模块[7]互连；SDRAM命令产生模块[5]、地址产生模块[7]分别和片外帧存储器连接；

片外帧存储器存贮了视频处理所需要的几帧图像，接口中的输入、输出缓存存在芯片内的处理部分与芯片外的SDRAM帧存储器之间形成数据交换接口，芯片外SDRAM的读写操作的发生是通过写请求模块[3]通过检测输入缓存的堆栈深度，产生写“呼叫”；写请求模块[4]的“呼叫”发生在输入缓存存储的数据要大于 $L_{burst}$ ；其中 $L_{burst}$ 表示读出或读入SDRAM中的数据个数；同理，读请求模块[3]通过检测输出缓存[2]的堆栈深度，产生读“呼叫”；读请求模块[3]的“呼叫”发生在输入缓存存储的数据要小于 $(L_{buffer} - L_{burst})$ ，其中 $L_{buffer}$ 表示输入和输出缓存的长度；仲裁模块[6]通过对读、写“呼叫”进行仲裁，产生相应的允许读写信号，激励SDRAM命令产生模块和地址产生模块产生相应的SDRAM命令和地址。

2. 实现权利要求1所述的高速视频处理接口控制器的处理方法，其特征在于，按以下步骤进行：

(1) 输入视频数据以固定的速率进入输入缓存，当输入缓存中的数据堆栈到一个阈值时，写请求模块的功能产生写请求信号；“呼叫”发生在输入缓存存储的数据要大于 $L_{burst}$ ；其中 $L_{burst}$ 表示读出或读入SDRAM中的数据个数；

(2) 输出缓存以固定的速率输出数据，当输出缓存中的数据低于某一阈值时读请求模块的功能产生读请求信号；“呼叫”发生在输入缓存存储的数据要小

于( $L_{\text{buffer}} - L_{\text{burst}}$ ), 其中  $L_{\text{buffer}}$  表示输入和输出缓存的长度;

(3) 仲裁模块当接收到写请求模块和读请求模块的请求后, 对其进行仲裁, 确定目前是进行 SDRAM 的写操作还是读操作, 并产生相应的地址触发信号, 使得地址产生器产生相应的读写地址, 这些读写地址是一种相对地址, 通过地址映射成为 SDRAM 的实际的物理地址;

(4) 当仲裁模块发出写命令后, 命令产生器产生 SDRAM 写操作, 数据从输入缓存中读出, 写入 SDRAM;

(5) 当仲裁模块发出读命令后, 命令产生器产生 SDRAM 读操作, 数据从 SDRAM 中读出, 写入输出缓存。

3. 如权利要求 2 所述的高速视频处理接口控制器的处理方法, 其特征在于, 写请求模块通过检测输入缓存的堆栈深度, 产生写“呼叫”; 同理, 读请求模块通过检测输出缓存的堆栈深度, 产生读“呼叫”; 仲裁模块通过对读、写“呼叫”进行仲裁, 产生相应的允许读写信号, 激励 SDRAM 命令产生模块和地址产生模块产生相应的 SDRAM 命令和地址; 则:

(1)、如果读呼叫请求有效并且写呼叫请求无效, 那么仲裁器发出允许读激励信号, 并把下一次总线状态交给“写”, 设置标志信号下一状态为 1, 表示写;

(2)、如果写呼叫请求有效并且读呼叫请求无效, 那么仲裁器发出允许写激励信号, 并把下一次总线状态交给“读”, 设置标志信号下一状态为 0。表示读;

(3)、如果请求信号读呼叫请求和写呼叫请求同时有效, 那么 SDRAM 仲裁器需要根据下一状态的状态来授权; 下一状态=1 时, 仲裁器发出允许写激励信号, 并把下一状态置成 0; 下一状态=0 时, 仲裁器发出允许读激励信号, 并把下一状态置成 1。

## 高速视频处理接口控制器及其处理方法

### 一、所属技术领域

本发明属于接口控制器技术领域，特别涉及到高速视频处理接口控制器其处理方法。

### 二、背景技术

电视对于当今世界任何国家来说，都是最重要的消费电子产品。数字化、网络化是电视发展的方向。近年技术上的进展，特别是数字编码、数字传输、集成电路和显示器件等方面的实用技术突破，已在逐步形成世界范围内的数字化处理电视市场。数字化处理电视与现行电视相比，其水平和垂直两个方向的图像分辨率都有一定提高，使用大屏幕显示器近距离观看时，图像细腻逼真，无闪烁和粗糙感，与模拟电视相比，数字化处理电视有以下优点：1.收视效果好，图像清晰度高音频质量高，满足人们感官的需求。2.抗干扰能力强。3.兼容现有模拟电视机。4.提供全新业务。

在数字化处理电视当中需要对大量的视频数据进行处理。视频处理也是目前多媒体应用领域的一个关键技术，主要包括视频信号的转换，视频数据的压缩、传输，以及视频信号的输入/输出等一系列重要课题。视频数据的采集是视频信号处理的第一步。视频处理的数据源通常为模拟电视信号，如 NTSC 或者 PAL 电视信号。对这些复合电视信号进行采样、亮色分离、同步产生等输出符合标准的数字化信号如 CCIR601, CCIR656 信号后，才能对这些信号进行数字化处理。在得到数字视频信号后，还需要将它们采集到视频处理系统的存储器中，视频处理系统才能对数据进行处理。

用于视频数据流的数据量大，实时性要求高，需要高速大容量的存储器作为图像数据的缓存。SDRAM 相比于 SRAM 等存储器具有容量大、速度快等优点，因此成为图像处理中常用的数据存储器。但是，SDRAM 控制较复杂，需要

处理换行、刷新等操作，因此需要设计 SDRAM 控制器以完成和 SDRAM 的接口。视频接口模块完成同步处理后将有效数据交给 SDRAM 控制器存入 SDRAM 中。视频处理器也通过 SDRAM 控制器和 SDRAM 接口，对视频数据进行压缩等处理后送至传输接口或者视频 D/A。可见，SDRAM 控制器的设计部分是其中的关键部分，它的性能直接影响到整个系统的性能。

根据申请人所作的资料检索，没有查到与本课题有关的文献。

### 三、 发明内容

本发明的目的在于，提供一种高速视频处理接口控制器其处理方法，本发明采取新的存储器接口策略，以达到改善视频处理中存储器的带宽和降低功耗的目的。采用 SDRAM（同步 DRAM）作为芯片外部的帧存储器，并针对 DTV、HDTV 实时视频处理芯片中的运动估计类算法设计应用，研究了一种帧存储器接口体系结构及称之为“被动呼唤”的设计策略，目的改善视频处理带宽，减少片上存储器面积，降低整体功耗。

为了实现上述目的，本发明采用的技术方案是，高速视频处理接口控制器由输入缓存模块、输出缓存模块，读请求模块，写请求模块，SDRAM 命令产生模块，仲裁模块，地址产生模块及地址映射模块组成；输入缓存模块与写请求模块连通，输出缓存模块与读请求模块连通，并分别与仲裁模块相连，仲裁模块分别与相互连接的 SDRAM 命令产生模块、地址产生模块互连；SDRAM 命令产生模块、地址产生模块分别和片外帧存储器连接；

片外帧存储器存贮了视频处理所需要的几帧图像，接口中的输入、输出缓存在芯片内的处理部分与芯片外的 SDRAM 帧存储器之间形成数据交换接口，芯片外 SDRAM 的读写操作的发生是通过写请求模块通过检测输入缓存的堆栈深度，产生写“呼叫”；写请求模块的“呼叫”发生在输入缓存存储的数据要大于  $L_{burst}$ ；其中  $L_{burst}$  表示读出或读入 SDRAM 中的数据个数；同理，读请求模块[3]通过检测输出缓存[2]的堆栈深度，产生读“呼叫”；读请求模块的“呼叫”发

生在输入缓存存储的数据要小于 $(L_{buffer} - L_{burst})$ ，其中  $L_{buffer}$  表示输入和输出缓存的长度；仲裁模块通过对读、写“呼叫”进行仲裁，产生相应的允许读写信号，激励 SDRAM 命令产生模块和地址产生模块产生相应的 SDRAM 命令和地址。

实现上述高速视频处理接口控制器的设计方法，按以下步骤进行：

(1) 输入视频数据以固定的速率进入输入缓存，当输入缓存中的数据堆栈到一个阈值时，写请求模块的功能产生写请求信号；“呼叫”发生在输入缓存存储的数据要大于  $L_{burst}$ ；其中  $L_{burst}$  表示读出或读入 SDRAM 中的数据个数；

(2) 输出缓存以固定的速率输出数据，当输出缓存中的数据低于某一阈值时读请求模块的功能产生读请求信号；“呼叫”发生在输入缓存存储的数据要小于 $(L_{buffer} - L_{burst})$ ，其中  $L_{buffer}$  表示输入和输出缓存的长度；

(3) 仲裁模块当接收到写请求模块和读请求模块的请求后，对其进行仲裁，确定目前是进行 SDRAM 的写操作还是读操作，并产生相应的地址触发信号，使得地址产生器产生相应的读写地址，这些读写地址是一种相对地址，通过地址映射成为 SDRAM 的实际的物理地址；

(4) 当仲裁模块发出写命令后，命令产生器产生 SDRAM 写操作，数据从输入缓存中读出，写入 SDRAM；

(5) 当仲裁模块发出读命令后，命令产生器产生 SDRAM 读操作，数据从 SDRAM 中读出，写入输出缓存。

写请求模块通过检测输入缓存的堆栈深度，产生写“呼叫”；同理，读请求模块通过检测输出缓存的堆栈深度，产生读“呼叫”；仲裁模块通过对读、写“呼叫”进行仲裁，产生相应的允许读写信号，激励 SDRAM 命令产生模块和地址产生模块产生相应的 SDRAM 命令和地址；则：

①如果读呼叫请求有效并且写呼叫请求无效，那么仲裁器发出允许读激励信号，并把下一次总线状态交给“写”，设置标志信号下一状态为 1，表示写；

②如果写呼叫请求有效并且读呼叫请求无效，那么仲裁器发出允许写激励

信号，并把下一次总线状态交给“读”，设置标志信号下一状态为0。表示读；

③如果请求信号读呼叫请求和写呼叫请求同时有效，那么 SDRAM 仲裁器需要根据下一状态的状态来授权；下一状态=1 时，仲裁器发出允许写激励信号，并把下一状态置成 0；下一状态=0 时，仲裁器发出允许读激励信号，并把下一状态置成 1；

本发明在运动估计的算法实现中：（1）只与每一执行的时间循环同步，即同步于每一新帧的开始；（2）数据输入和输出的次序一定且不变；（3）在同一时间内需要得到 2 帧的视频数据；采用本发明能够使帧存储器存取功耗的降低。

#### 四、附图说明

图 1.1 是帧存储器接口体系具体结构图；

图 1.2 是帧存储器接口体系结构示意图。

#### 五、具体实施方式

以下结合附图对本发明作进一步的详细描述。

1) 本发明的帧存储器接口体系结构如图 1.1 所示。图中虚线中部分为帧存储器接口，由输入缓存 (Input Buffer)、输出缓存 (Output Buffer)，读请求 (Read Beg)，写请求 (Write Beg)，SDRAM 命令产生 (SDRAM Command Gnerator)，仲裁 (Arbitrate)，地址产生 (Address Generator) 及地址映射 (Address Map) 几部分功能模块组成。输入缓存模块 1 与写请求模块 3 连通，输出缓存模块 2 与读请求模块 4 连通，并分别与仲裁模块 6 相连，仲裁模块 6 分别与相互连接的 SDRAM 命令产生模块 5、地址产生模块 7 互连；SDRAM 命令产生模块 5、地址产生模块 7 分别和片外帧存储器连接。

片外帧存储器存贮了视频处理所需要的几帧图像，接口中的输入、输出缓存在芯片内的处理部分与芯片外的 SDRAM 帧存储器之间形成数据交换接口，芯

片外 SDRAM 的读写操作的发生是通过一种称之为“被动呼唤”的设计策略完成的。

2) 设在运动估计的算法实现中：(1) 只与每一执行的时间循环同步，即同步于每一新帧的开始；(2) 数据输入和输出的次序一定且不变；(3) 在同一时间内需要得到 2 帧的视频数据。由于在 DTV、HDTV 实时视频后处理芯片中的运动估计算法，与用于视频压缩的运动估计算法不同，其输出帧频要大于输入帧频，且有：

$$f_{output} = nf_{input} \quad (1.1)$$

其中  $f_{input}$  和  $f_{output}$  分别表示视频信号的输入输出频率， $n \geq 2$ 。则在每一写入新帧的时间循环中，帧存储器接口会完成对 SDRAM 的一帧写操作， $2n$  帧读操作。定义  $\sigma$  为每一时间循环中的同步间隔的和，则有：

$$\sigma = T_{input} - T_{sdram\_write} - 2nT_{sdram\_read} \quad (1.2)$$

式(1.2)中， $T_{input}$  表示输入每一新帧所需要的时间周期， $T_{sdram\_write}$  和  $T_{sdram\_read}$  分别表示向 SDRAM 写入一帧和从 SDRAM 读出一帧视频数据所需要的时间周期。理论上， $\sigma$  越小表示增加的同步间隔越少， $\sigma$  若为零，则可表示为理想同步。

由图 1.1 可知，芯片的输入和输出是以固定的速率进行的，因此，就要求帧存储器接口中的输入缓存的输入和输出缓存的输出要有固定的速率。因此，一方面为减少片上缓存的面积，尽可能使得  $\sigma$  趋近于零；另一方面，为使得处理与 I/O 流保持完全的同步，需要增加同步间隔，也即增加  $\sigma$  来保持同步。

因此，本发明提出一种称之为“被动呼唤”的帧存储器接口设计策略，其原理如下：如图 1.1 所示，写请求模块通过检测输入缓存的堆栈深度，产生写“呼叫”；同理，读请求模块通过检测输出缓存的堆栈深度，产生读“呼叫”；仲裁模块通过对读、写“呼叫”进行仲裁，产生相应的允许读写信号，激励 SDRAM



命令产生模块和地址产生模块产生相应的 SDRAM 命令和地址。图 1.2 给出了仲裁模块的最基本的状态机流程图。图中 read request 和 write request 分别表示读、写“呼叫”请求，则：

(1) 如果 read request 有效并且 write request 无效，那么仲裁器发出允许读激励信号，并把下一次总线状态交给“写”，设置标志信号 next\_state 为 1(表示写)。

(2) 如果 write request 有效并且 read request 无效，那么仲裁器发出允许写激励信号，并把下一次总线状态交给“读”，设置标志信号 next\_state 为 0(表示读)。

(3) 如果请求信号 read request 和 write request 同时有效，那么 SDRAM 仲裁器需要根据 next\_state 的状态来授权。next\_state=1 时，仲裁器发出允许写激励信号，并把 next\_state 置成 0；next\_state=0 时，仲裁器发出允许读激励信号，并把 next\_state 置成 1。

由此可知，仲裁模块会根据读、写“呼叫”请求，均匀分配 SDRAM 读写操作。这一方面使得输入与输出缓存的大小可以一样，利于减小同步间隔 $\sigma$ ；另一方面，可使 SDRAM 的读写操作在较小的 Burst 长度下，使得处理与 I/O 流易于保持完全的同步，满足同一时间内需要得到多个帧的视频数据这个条件。下面我们将讨论这种帧存储器接口设计对片上存储器面积、视频带宽及功耗的影响。

3) 设 SDRAM 的 Burst 存取模式下，Burst 长度为  $L_{Burst}$  个时钟周期。根据图 1.1 的体系结构，帧存储器接口每完成一次 SDRAM 写操作，就会从输入缓存读出  $L_{Burst}$  个数据写入片外 SDRAM；同理，帧存储器接口每完成一次 SDRAM 读操作，就会从片外 SDRAM 读入  $L_{Burst}$  个数据写入输出缓存。考虑到输入缓存的输入和输出缓存的输出是以固定的速率输入输出的，就使得写请求模块的“呼叫”发生在输入缓存存贮的数据要大于  $L_{Burst}$ ；读请求模块的“呼叫”发生在输出缓存存贮的数据小于  $(L_{buffer} - L_{Burst})$ ，其中  $L_{buffer}$  表示输入和输出缓存的长度。由式(1.1)可知，输出的帧频大于等于 2 倍的输入帧频，引入式(1.1)中的  $n$ ，则可确定  $L_{buffer}$  的大小为：

$$L_{buffer} \geq 2(n+2)L_{Burst} \quad (1.3)$$

式(1.3)说明, 芯片内部的缓存大小与 Burst 长度  $L_{Burst}$  成正比。因此, 要减少芯片内部的缓存大小, 只需选取较小的 Burst 长度。

考虑芯片与 SDRAM 帧存储器之间的视频带宽与 Burst 长度的关系。设  $L_{write\_head}$  和  $L_{read\_head}$  表示在 Burst 模式下, 使读写行列地址有效的时钟周期的数目, 即额定开销的时钟周期。设  $T_{write\_once}$ 、 $T_{read\_once}$  为进行一次 SDRAM 写操作或读操作所必须的时间,  $f_{sdram}$  为 SDRAM 工作时钟频率, 则有:

$$T_{write\_once} = (L_{write\_head} + L_{Burst}) / f_{sdram} \quad (1.4)$$

$$T_{read\_once} = (L_{read\_head} + L_{Burst}) / f_{sdram} \quad (1.5)$$

对于图像大小为  $W \times H$  的视频输入帧来说, 帧存储器接口向 SDRAM 写入一帧和从 SDRAM 读出一帧视频数据所需要的时间为:

$$T_{write\_one\_frame} = \frac{W \times H}{L_{Burst}} T_{write\_once} \quad (1.6)$$

$$T_{read\_one\_frame} = \frac{W \times H}{L_{Burst}} T_{read\_once} \quad (1.7)$$

由上述公式, SDRAM 读写一帧视频数据所需要的额定时间开销  $T_{write\_overhead}$  和  $T_{read\_overhead}$  分别为:

$$T_{write\_overhead} = \frac{W \times H}{L_{Burst} \cdot f_{sdram}} L_{write\_head} \quad (1.8)$$

$$T_{read\_overhead} = \frac{W \times H}{L_{Burst} \cdot f_{sdram}} L_{read\_head} \quad (1.9)$$

在确定了 SDRAM 的操作方式后,  $L_{write\_head}$ 、 $L_{read\_head}$  和  $f_{sdram}$  就可确定, 式(1.8)和(1.9)说明 SDRAM 读写的额定时间开销与  $L_{Burst}$  成反比。由于读写相同  $L_{Burst}$  的视频数据所需要的额定时间开销越大, 视频带宽越小, 也即视频带宽与  $L_{Burst}$  成正

比；因此，要增大视频带宽，就需加大  $L_{Burst}$ 。

因此，在确定了片上存储器面积后，对视频带宽进行改善。具体策略如下：

策略 1：在选取较小的 Burst 长度情况下，在图 1.1 中的处理模块中加入排队处理，使输入输出缓存的数据宽度增加一倍，也即与式 (1.3) 相比，使  $L_{buffer}$  在容量上增加一倍为：

$$L_{buffer} \geq 4(n+2)L_{Burst} \quad (1.10)$$

这也使得帧存储器接口向 SDRAM 写入一帧和从 SDRAM 读出一帧视频数据所需要的时间  $T'_{write\_one\_frame}$  和  $T'_{read\_one\_frame}$  与式 (1.6) 和 (1.7) 中的  $T_{write\_one\_frame}$  和  $T_{read\_one\_frame}$  相比，有：

$$T'_{write\_one\_frame} = T_{write\_one\_frame} / 2 \quad (1.11)$$

$$T'_{read\_one\_frame} = T_{read\_one\_frame} / 2 \quad (1.12)$$

策略 2：在图 1.1 中的地址映射模块，对 SDRAM 的地址进行变换，使得运动估计的当前帧与前一帧的数据存贮在 SDRAM 中的不同 BANK 中，利用 SDRAM 的特征—多 Bank 体系结构，则可在同一时间内，在不同的视频帧中读出相同 Burst 长度的视频数据，使得进行一次 SDRAM 读操作所必须的时间  $T_{read\_once}$  为：

$$T_{read\_once} = (L_{read\_head} + 2L_{Burst}) / f_{sdram} \quad (1.13)$$

从 SDRAM 读出两帧视频数据所需要的时间为：

$$T_{sdram\_read\_two} = \frac{W \times H}{L_{Burst} \cdot f_{sdram}} (L_{read\_head} + 2L_{Burst}) \quad (1.14)$$

在读出两帧视频数据的相同情况下，改进后的方法比用式 (1.7) 的原方法所减少了的时间  $T_{cut\_down}$  为：

$$T_{cut\_down} = \frac{W \times H}{L_{Burst} \cdot f_{sdram}} \left( \frac{3}{2} L_{read\_head} + L_{Burst} \right) \quad (1.15)$$

式 (1.15) 说明, 通过上述的策略, 很好地解决了减少片上存储器面积与改善视频带宽这一对矛盾。在确定了  $L_{buffer}$  的容量为  $4(n+2)L_{Burst}$  后, 芯片与外部 SDRAM 帧存储器之间的视频带宽, 得到明显改善。

(4) 片上存储器面积的减少, 肯定会带来芯片功耗的降低。现在, 我们来分析这种帧存储器接口设计带来的存储器存取功耗降低。如式 (1.16) 所示, 与存储器存取相关的功耗可以分为两个部分: 内核功耗, I/O 功耗;

$$P_{total} = P_{core} + P_{I/O} = I_{core}V_{dd} + \alpha CV_{dd}^2 f \quad (1.16)$$

其中,  $I_{core}$ ,  $\alpha$ ,  $C$  和  $f$  分别表示 SDRAM 平均电流, I/O 总线的平均开关率, I/O 电容及工作频率。当使用的 SDRAM 类型确定后,  $\alpha$ ,  $C$  和  $f$  即可确定。存储器存取相关的功耗的减少主要与  $I_{core}$  有关。若假设行有效、Burst 存取和刷新操作的平均电流分别为 150, 180 和 210mA, 则可得到简化的有关  $I_{core}$  的公式:

$$I_{core} = \frac{1}{T} (8.40 \times 10^{-9} \times CN + 1.35 \times 10^{-9} \times DN + 13.7 \times 10^{-9} \times RN) \quad (1.17)$$

$CN$ 、 $DN$  和  $RN$  分别表示在时间  $T$  内, 行有效的次数、列数据存取次数和刷新次数。由于刷新次数与前两项相比微不足道, 而一次行有效导致的电流是一次列数据存取的 6 倍, 因此, 存储器存取相关的功耗的减少主要与行有效的次数  $CN$  有关。

由式 (1.15), 我们可得到在从 SDRAM 读出两帧的视频数据的时间内, 本文所述的帧存储器接口设计有效降低行有效的次数约为:

$$CN_{cut\_down} \approx \frac{W \times H}{L_{Burst}} \left( \frac{3}{2} L_{read\_head} + L_{Burst} \right) \quad (1.18)$$

式 (1.18) 定量说明了本文所述的帧存储器接口设计所带来的存储器存取功耗的降低。

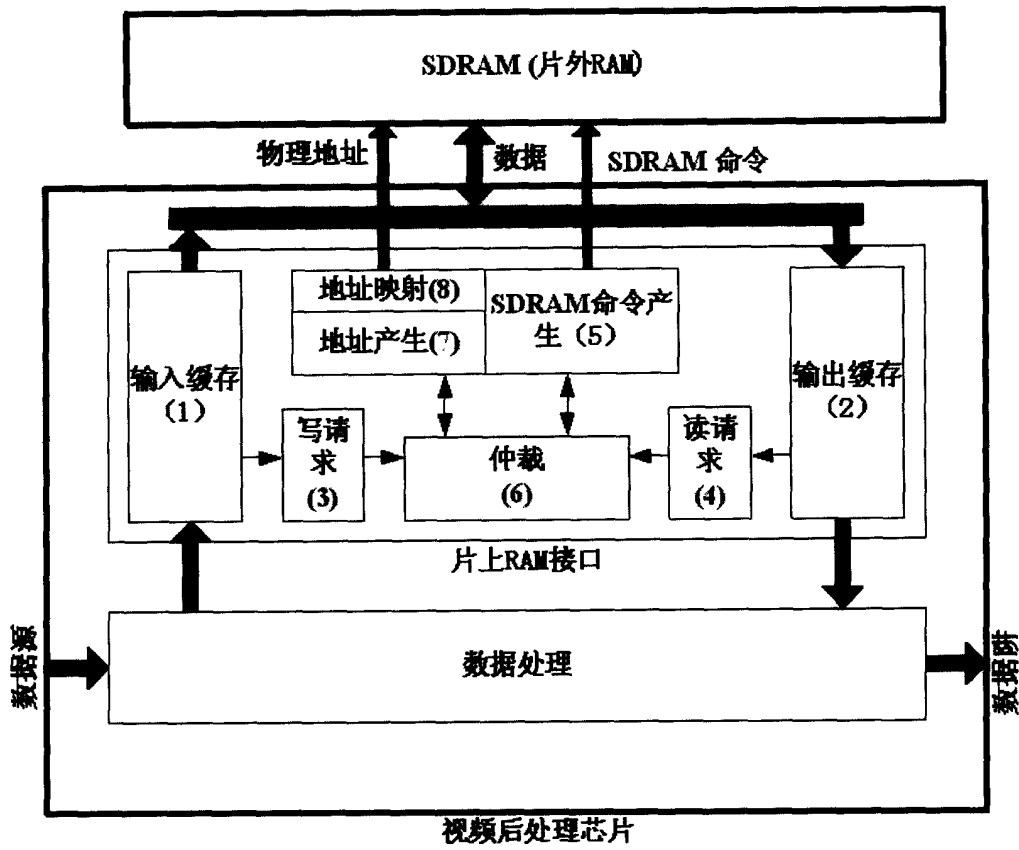


图 1.1

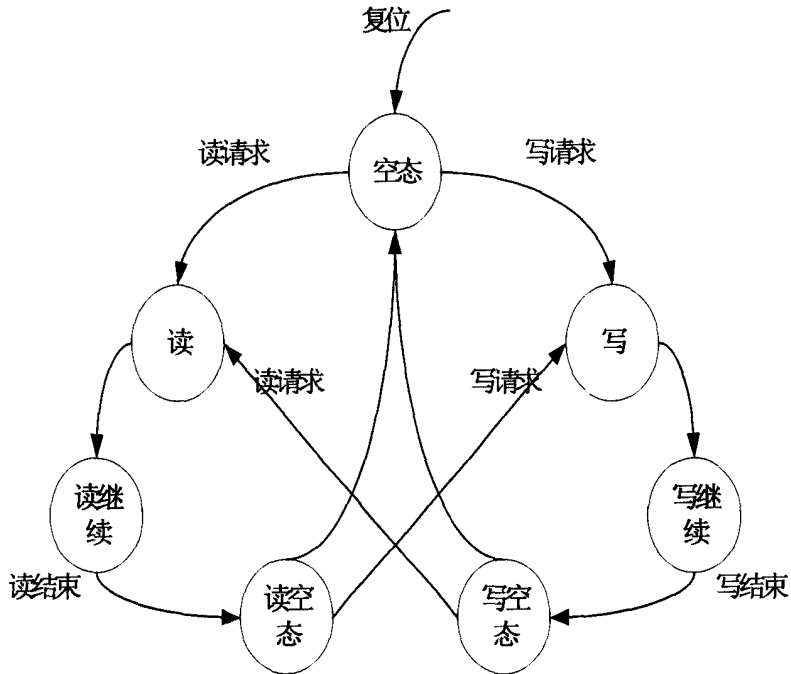


图 1.2