



[12] 发 明 专 利 说 明 书

[21] ZL 专利号 99110437.4

[45] 授权公告日 2004 年 7 月 7 日

[11] 授权公告号 CN 1157063C

[22] 申请日 1999. 7. 12 [21] 申请号 99110437.4

[30] 优先权

[32] 1998. 7. 13 [33] US [31] 09/114568

[71] 专利权人 摩托罗拉公司

地址 美国伊利诺斯

[72] 发明人 戴维·J·霍金斯

布鲁斯·A·奥古斯丁

审查员 苏 青

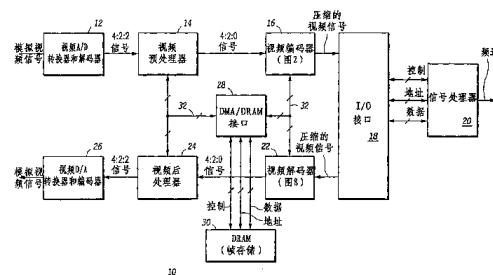
[74] 专利代理机构 中国国际贸易促进委员会专利
商标事务所
代理人 付建军

权利要求书 1 页 说明书 12 页 附图 8 页

[54] 发明名称 通过存储和检索运动矢量对视频信号进行编码和解码的方法和装置

[57] 摘要

一个能对符合压缩标准, 如 H. 263 标准的视频信号进行编码解码并能高效地存储和检索与信号有关的运动矢量的系统(10)。编码器(16)和解码器(22)各使用一个运动矢量存储和检索电路(38, 72), 该电路在各自的 SRAM 数组(52, 52')中寻址。通过除去当前计算不需要的运动矢量, 只存储最少量的运动矢量来实现视频信号的先行预测模式。避免存储预先确定的宏块的上运动矢量来节省内存。



1. 一种用于对视频信号进行编码和解码的存储和检索运动矢量的装置，该装置的特征在于：

有一组存储元素的内存，该组存储元素的第一个元素存储一个视频帧的一个宏块的第一个运动矢量，该宏块是表示一个视频帧的一组宏块中的一个，存储元素组的第二个元素存储宏块的第二个运动矢量，所存储的第一运动矢量和第二运动矢量都是同一个宏块的，第一个运动矢量和第二个运动矢量由运动矢量预测值计算和用于消除视频非自然信号的重叠块运动补偿所共享；和

与内存相连接的地址生成电路，该地址生成电路控制运动矢量存储位置；

其中，内存和地址生成电路都是由集成电路形式的硬件元件来实现。

2. 如权利要求 1 的装置，其特征还在于包括一个响应内存的预测值计算器。

3. 如权利要求 1 的装置，其中第一个运动矢量指向宏块的上半部分，第二个运动矢量指向宏块的下半部分，内存存储一组运动矢量，第一组运动矢量对应于至少一个宏块的上半部分，第二组运动矢量对应于至少一个宏块的下半部分，第一组运动矢量在内存中所占的存储元素少于第二组运动矢量。

通过存储和检索运动矢量对视频 信号进行编码和解码的方法和装置

技术领域

本发明一般涉及视频通信,具体地说,涉及视频图像的压缩和解压缩。

背景技术

由于现有电话网络固有的带宽限制,大多数商业用和家庭用的可视电话和电视会议设备要求很高的压缩比率。为了满足这种要求,开发了 H.263 低比特率视频通信国际标准。该标准很快就被视频工业采用。H.263 国际标准的优点包括一种称为先行预测模式 (APM) 的可选技术的实现。通过消除运动估计带来的非自然信号 (artifact), APM 明显地改善了视频图象。APM 的缺点是用硬件实现很困难而且效率低。因此,先行预测模式的首次实现是仅由软件来实现的。

压缩视频的一种 H.263 技术是采用运动估计。运动估计通过将运动矢量而不是真正的象素送给解码器来处理视频。解码器接收运动矢量并通过访问运动矢量所指向的数据块由上一个解码的帧生成当前帧。H.263 APM 模式中,传送一个运动矢量相当于传送一个 8×8 的象素块。APM 选择要求代表每个 8×8 块的运动矢量与那些代表四周块的矢量平均。这样可以减少由运动估计处理导致的分块非自然信号。不幸的是,出于几种原因从周围的宏块平均运动矢量需要大量的内存和硬件。首先,从周围的块来取得运动矢量需要在内存中存储大量的运动矢量。要取得当前块的运动矢量需要存储整整一行的运动矢量,这很耗费内存。第二, H.263 标准定义要求 APM 模式为两个独立功能存储和检索运动矢量:重叠块运动补偿 (OBMC) 和预测值计算。这些功能不能轻易地结合起来是因为 OBMC 功能滞后预测器功能一个宏块的解码时间,这就要求这些功能在不同的时间帧里操作。第三, OBMC 和预测值计算各要求四种独立的地址方式。因此要求有八种独立的寻址方式。但是,没有一种寻址方式是线性顺序,也没有一种寻址方式是容易编址的任何其它方式。

由于上述的问题,很难设计出可以高效地实现由 H.263 标准定义的 APM 模式的硬件用软件来实现 APM 模式的优越性在于可以获得实现这个功能所要求的大量的内存和系统资源。但是,用软件实现运行速度慢,比

起硬件实现降低了帧速率和图像质量。

发明内容

本发明提供了一种用于对视频信号进行编码和解码的存储和检索运动矢量的装置，该装置的特征在于：有一组存储元素的内存，该组存储元素的第一个元素存储一个视频帧的一个宏块的第一个运动矢量，该宏块是表示一个视频帧的一组宏块中的一个，存储元素组的第二个元素存储宏块的第二个运动矢量，所存储的第一运动矢量和第二运动矢量都是同一个宏块的，第一个运动矢量和第二个运动矢量由运动矢量预测值计算和用于消除视频非自然信号的重叠块运动补偿所共享；和与内存相连接地址生成电路，该地址生成电路控制运动矢量存储位置；其中，内存和地址生成电路都是由集成电路形式的硬件元件来实现。

附图说明

图 1 以方框图的形式示出了使用本发明的一个视频通信系统。

图 2 以方框图的形式说明了根据本发明的视频编码器。

图 3 以方框图的形式说明了根据本发明的图 2 的运动矢量存储和取得的电路图。

图 4 以透视图的形式说明了一个公用中间格式大小帧的一行的宏块的布局。

图 5 说明了符合 H.263 标准的预测值计算用的运动矢量检索模式。

图 6 说明符合 H.263 标准的 OBMC 功能用的运动矢量检索模式。

图 7 说明图 3SRAM 数组的内存分配。

图 8 以方框图形式说明了根据本发明的在图 1 所示的系统中使用的视频解码器的框图。

图 9 以框图的形式说明图 8 所示的解码功能的运动矢量存储和检索的电路图。

具体实施方式

图 1 所示为依照本发明的视频编解码器装置的框图。本发明克服了上述的问题，从而允许用硅来高效地实现 APM。视频 A-D 转换器和解码器 12 有一个用于接收模拟视频信号的输入端，用于产生 4: 2: 2 视频格式的信号的输出端。4: 2: 2 信号连接到视频预处理器的一个输入端，用预处理器产生 4: 2: 0 视频格式的信号的输出端。视频预处理器 14 的一个输出端连接到视频编码器 16 的一个输入端。视频编码器 16 将在图 2 中进一步详细说明。视频编码器 16 给输入/输出 (I/O) 接口电路 18 提供一个压缩视频位流。I/O 接口 18 通过控制、地址、数据总线与信号处理器 18 相连。I/O 接

口 18 的功能是为压缩视频位流和信号处理器 20 中所使用的特定类型的信号处理器提供接口。I/O 接口 18 还与视频解码器 22 的一个输入端相连，为其提供压缩的视频位流。视频解码器 22 的一个输出端与后处理器 24 的一个输入端相连。视频解码器 22 的输出是 4: 2: 0 视频格式。后处理器 24 的一个输出端以 4: 2: 2 的视频格式连到视频 D-A 转换器和编码器 26。视频 D-A 转换器和编码器 26 产生一个模拟视频信号的输出。图中还有直接内存访问 (DMA) / 动态随机访问内存 (DRAM) 接口电路 28。DMA/DRAM 接口电路 28 通过内部总线 32 分别连接到视频预处理器 14，视频编码器 16，视频解码器 22 和后处理器 24。DMA/DRAM 接口电路 28 通过控制、地址和数据总线与 DRAM30 相连接。视频编解码器 10 的功能是接收一个模拟视频信号，通过视频 A-D 转换器和解码器 12 和视频预处理器 14 将其转换为两种不同的视频格式，提供一个 4: 2: 0 信号给视频编码器，该视频编码器将信号压缩为压缩的视频位流。压缩的视频位流送到信号处理器 20 进行处理，信号预处理器 20 就象一个主机将压缩的视频位流提供给发送频道。根据所使用频道的类型，信号处理器可能把压缩视频信号转换为模拟格式或保持其数据格式。I/O 接口 18 也提供压缩视频位流给视频解码器 22，该压缩视频位流是通过信号处理器 20 从频道接收到的。视频解码器 22 将视频位流解压缩为 4: 2: 0 视频格式交由后处理器 24 使用。后处理器 24 将 4: 2: 0 视频格式转换为视频 D/A 转换器 26 使用的 4: 2: 2 格式。视频 D/A 转换器提供一个可由传统的电视或视频监视器接收的模拟输出。视频编解码器 10 的构成如图 1 所示是很传统的。换句话说，用现有的集成电路产品可以实现视频 A-D 转换器和解码器 12，视频预处理器 14，后处理器 24 和视频 D-A 转换器和编码器 26。但是，应用不同视频编码器 16 和视频解码器 22 的实现和效率也将不同。本发明涉及视频编码器 16 和视频解码器 22 的特殊实现方法。

图 2 是图 1 的视频编码器 16 的详细框图。视频编码器 16 一般包括一个运动估计器电路 36，该电路有一个接收 4: 2: 0 视频格式信号的输入端。运动估计器 36 提供很多运动矢量给运动矢量存储和检索模块 38。模块 38 的一个输出端与运动补偿和位移帧差电路 42 的一个输入端相连接。运动补偿和位移帧差 (displaced frame difference) 电路 42 的一个输出端连接到离散余弦变换和反离散余弦变换 (IDT/DCT) 电路 44 或模块的一个输入端。一般来说，IDT/DCT 电路 44 基于压缩/解压缩算法 (例如小波 (wavelet) 或离散余弦变换) 来实施变换。离散余弦变换和反 DCT 电路 44 的一个输出端连接到位流解码器 46 的一个输入端，位流解码器 46

产生压缩的视频输出位流。比率控制电路 (rate control circuit) 48 通过图 1 中也示出的内部总线 32, 分别连接到运动估计器 36、运动补偿和位移帧差电路 42 和离散余弦变换和反 DCT 电路 44。

在工作中, 视频编码器 16 的作用是根据上一个像素帧的信息对当前帧进行运动估计。运动估计器 36 产生多个运动矢量。运动矢量表示在前面帧和当前帧之间的 8×8 像素块的位移。运动矢量存储和检索模块 38 接收运动矢量。正如将在图 3 中详细说明的那样, 模块 38 的功能是检索和存储用于视频标准如 H.263 国际标准中的先行预测模式所要求的运动矢量。一旦模块 38 检索和存储了这样的运动矢量, 模块 38 将在它的输出端产生两种运动矢量给运动补偿或位移帧差电路 42。第一种运动矢量用于重叠块运动补偿。第二种运动矢量用于重建前一帧和运动补偿。运动补偿和位移帧差电路 42 接收这两种运动矢量并完成两个功能: 位移帧差和运动补偿。运动补偿功能使用上述的两种运动矢量重复远程解码器的解码。运动补偿功能通过总线 32 为运动估计器 36 提供一个重建参考帧。这个帧也将被引用为前一帧或重建帧, 是将被远程解码器解码的帧的精确复制。当视频编码器编码下一帧时, 它将被运动估计器 36 用于运动估计的参考帧。位移帧差功能从原始 (参考) 帧中减去运动估计帧并提供其差给离散余弦变换和反 DCT 电路 44。离散余弦变换和反 DCT 电路 44 基于压缩函数进行有损变换 (lossy transform)。DCT/IDCT 44 的输出和来自运动估计器 36 的运动矢量提供给位流编码器 46。位流编码器 46 将数据排列为 H.263 标准所要求的格式, 并产生压缩视频位流输出到频道。除了通过频道被送出, DCT/IDCT 电路 44 的压缩的输出通过该 DCT/IDCT 电路 44 的反变换功能得以解压缩。解压缩后的输出, 通过总线 32, 送入到同一个运动补偿和位移差电路 42。这个信息将由位移帧差功能用于撤消 (undo) 它的上一次减法并重新产生一个运动估计器 36 的参考帧, 该帧正是在频道另一端的远程解码器将要解码的那一帧。

图 3 所示为依照本发明的图 2 的运动矢量存储和检索模块 38。运动矢量寄存器 50 的输出提供当前的运动矢量给 SRAM 运动矢量数组 52, 该数组一般是指一组存储运动矢量的存储元素。基本地址计数器 54 用于产生一个基址, 与地址产生器 56 相连接。地址产生器 56 的第二个输入

端用于接收来自查找表 58 的偏移值。查找表 58 的第一个输入端用于接收块号。块号是在大多数视频标准中定义的宏块中的 4 个可能的块之一。查找表 58 的第二个输入端用于 OBMC/预测模式信号, 该信号用于实现 H.263 国际标准所必需的 OBMC 或预测值计算模式。查找表 58 的第三个输入端用于选择需要的运动矢量(左, 上, 右上, 右, 下), 这将在图 5 和图 6 中进一步说明。运动矢量的选择一般是由一个两位的状态机(未图示)来产生。状态机在 4 种可能的选择(左, 上, 右上, 右, 下)之间进行循环并在开始一个新宏块的编码时复位回到开始。模加法器 57 的第一个输入端从查找表 58 接收一个偏移值, 第三个输入端从基址计数器 54 接收一个基址, 它的输出端产生一个地址, 连接到 SRAM 运动矢量数组 52 上。SRAM 数组 52 的输出端提供选中的运动矢量给多路复用器电路 60。多路复用器 60 的第一个输出端与预测值计算器电路 62 的一个输入端相连接。多路复用器 60 的功能是响应 OBMC/预测模式控制信号, 该信号也用于从查找表 58 中选择正确的偏移量。多路复用器 60 的第二个输出端与运动补偿单元 42 的一个输入端相连接。运动补偿单元 42 的输出端提供信号给图 2 中所示的 DCT/IDCT 电路 44。在说明运动矢量存储和检索模块 38 之前, 先来参看一下图 4 到图 6。

图 4 是在一个共用中间格式(CIF)类型的视频帧中的一行的宏块的布局。宏块布局中有多行宏块。为了便于说明, 只详细列出一行宏块。每一行有 22 个宏块, 分别标识为 C0 到 C21。在图中, 宏块 C0 按时间代表当前宏块, 宏块 C0 的紧上方是宏块 C22, 在时间域里, 宏块 C22 是过去的第 22 个宏块。为了尽可能地节省硬件和内存, 必须共享和结合 OBMC 和预测器功能使用的内存。OBMC 功能必须比预测器功能滞后一个宏块的时间。这个因素致使 OBMC 功能在宏块 C-1 执行(存储和检索运动矢量), 而预测器功能则在宏块 C0 执行。必须将 OBMC 功能滞后在宏块 C1 是因为 OBMC 要求检查当前 OBMC 功能执行的宏块右边的运动矢量(这将在图 6 中进一步说明)。如果 OBMC 功能在宏块 C0 上执行, 那么“右边”的运动矢量将从宏块 C+1 中检索, 而该宏块 C+1 是未来的一个宏块, 现在还不能使用。在图 5 和图 6 中将进一步说明, 必须从 OBMC 和预测器功能的上边的宏块中检索运动矢量。且需要注意

的是，“上边”的运动矢量对 OBMC 功能来说是 22 个宏块前处理的，而对预测器功能来说是 23 个宏块前处理的。因此，保存 24 个（C0 到 C23）宏块是必要的。

在图 5 和图 6 中将用到以下术语：大方块是指一个 16×16 像素的宏块，可划分为 4 个 8×8 的像素块。每个 8×8 像素块都有自己的运动矢量。4 个 8×8 的像素块，在本技术领域通常称为“块”被编号为块 1，2，3 和 4。块 1 是宏块的左上角，块 2 是右上角，块 3 是左下角，块 4 是右下角。处理一个宏块是指对该宏块进行编码或解码。图 5 和图 6 中的术语 C-1 是指处理过的上一个宏块。依此类推，宏块 C21 是指过去处理的第 21 个宏块。“C0”是指当前正在处理的宏块。在图 5 中需要注意是在宏块中需要为块 1 检索的运动矢量是左、上、右上。左运动矢量来自宏块 C-1 的块 2。上运动矢量来自宏块 C-22 的块 3。右上运动矢量来自宏块 C-21 的块 3。当前处理宏块的块 2 的检索模式如下：左运动矢量是当前被处理的宏块（C0）中的块 1。上运动矢量是宏块 C-22 中的块 4。右上运动矢量是宏块 C-21 中的块 3。当前处理的宏块中的块 3 的检索模式也与块 1，块 2，块 3，块 4 不同。当前宏块中的块 3 的检索模式如下：左运动矢量是宏块 C-1 中的块 4。上运动矢量是当前处理的宏块（C0）的块 1。右上运动矢量是当前处理的宏块（C0）中的块 2。对当前处理的宏块的块 4 来说，左运动矢量是当前宏块 C0 中的块 3。上运动矢量是当前宏块 C0 中的块 1。右上运动矢量是当前宏块 C0 中的块 2。需要重申的是，宏块 1，2，3，4 的检索模式互不相同。这些模式定义在 H.263 标准中。

本发明为 OBMC 功能支持运动矢量的检索。这个功能也是对宏块的每一个块都有各自的检索模式。OBMC 功能要求为宏块中的每一个块检索左，上，右，下的运动矢量。需要重申的是，向右检索运动矢量比较困难，因为当前宏块右边的运动矢量还未被处理。为了检索右运动矢量，必须将 OBMC 功能在时间上滞后一个宏块。因此 OBMC 功能将在宏块 C-1 执行，比预测器功能滞后一个宏块。OBMC 功能的检索模式如下：对块 1 来说，左运动矢量是宏块 C-2 的块 2。上运动矢量是宏块 C-23 的块 3。右运动矢量是宏块 C-1 的块 2 下运动矢量是宏块 C-1

的块 3。对当前 OBMC 正在处理的宏块中的块 2 来说, 左运动矢量是宏块 C-1 中的块 1, 上运动矢量是宏块中的块 4。右运动矢量是宏块 C0 中的块 1, 下运动矢量是宏块 C-1 的块 4。对 OBMC 功能的块 3 来说, 左运动矢量是宏块 C-2 的块 4。上运动矢量是宏块 C-1 的块 1。右运动矢量是宏块 C-1 的块 4。下运动矢量是宏块 C-1 的块 3。对 OBMC 宏块 C-1 的块 4 来说, 左运动矢量是宏块 C-1 的块 3。上运动矢量是宏块 C-1 的块 2。右运动矢量是宏块 C0 的块 3, 下运动矢量为宏块 C-1 的块 4。

参考前面的图 3, 运动矢量存储和检索模块 38 的功能如下。OBMC 和预测模式的八种寻址方式的偏移量, 如图 5 和图 6 所示, 是由查找表 58 提供的。查找表 58 的输入包括在宏块中的编号 (1, 2, 3, 4), 所需的运动矢量 (左、右、等) 及为运动矢量检索选择正确偏移的功能模式 (OBMC/预测)。

要实现的第一种模式是预测模式。为了实现预测模式, 请参看图 5。图 5 示出了依照 H.263 国际标准定义的 4 个宏块组织。每个宏块内有四个表示 8×8 块的运动矢量。对于预测模式, H.263 国际标准要求宏块内的每一块检索左、上、右上的运动矢量。宏块内不同的块的检索左, 上, 右上运动矢量的检索模式不同。为了实现检索模式, 计数器 54 为地址生成器 56 提供一个 0 到 53 的数字。地址生成器 56 对计数器的输出乘以 2 以获得一个基址。在实现中必须乘以 2 是因为每个数中存储两个运动矢量。查找表 58 根据块号模式和运动矢量选择生成一个偏移值。模加法器 57 将查找表 58 提供的偏移值和计数器 54 提供的基址相加。其和是一个地址, 在与 SRAM 数组 52 通讯时作为一个指针。该地址用于依照 H.263 标准的检索模式检索左、上、右上运动矢量。在 SRAM 数组 52 中编址的左、上、右上运动矢量可通过多路复用器 60 (Mux) 由 SRAM 数组 52 选择性地输出。选中的运动矢量提供给预测值计算器 62。预测值计算器 62 的功能是计算选中的上、右上和左运动矢量的中值。计算所得的中值, 即所谓的“预测值”提供给位流编码器 46。在位流编码器 46 中, 预测值从运动估计器 36 提供的运动矢量中抽取出来。其结果, 在本技术领域称为“差”, 被送往频道。预测值计算器 62 的实现方法广为本技术领域人

员所知。

在 OBMC 模式下，查找表 58 根据 H.263 标准 OBMC 的运动矢量检索模式（图 6）提供一个偏移值。地址生成器 56 的功能与前面有关预测值计算的说明类似，将一个地址提供给 SRAM 数组 52，然后 SRAM 数组 52 提供选中的运动矢量给多路复用器 60。多路复用器 60 提供的 OBMC 运动矢量通过多路复用器 60 提供给运动补偿单元 42。运动补偿单元用 OBMC 运动矢量为运动估计器作成重建帧。

需要理解的是注意到不必存储宏块 C-2 以前的宏块的上运动矢量即块 1 和 2，因而可以大大地节省内存。请参照图 6 的块 1（OBMC 功能）来理解这一点。从图 6 可以看到最后一次使用上（T-X）运动矢量是在为宏块 C-1 的块 1 检索左运动矢量的时候。因此，宏块 C-2 之后只需要下半部分（B-X）运动矢量。这是因为为了获得“上”运动矢量只需要当前正在处理的宏块上面的宏块的下运动矢量。

图 7 所示为一种 SRAM 数组 52 内存分配一个例子。在图示中，有 27 个半个宏块，每半个宏块包括宏块的上半部分或下半部分的运动矢量。在内存布局中，有 24 个宏块的下半部分和 3 个宏块的上半部分。宏块的上半部分存储宏块的块 1 和块 2。宏块的下半部分存储宏块的块 3 和块 4。具体说来，分配给下运动矢量 48 个地址位置，分配给上运动矢量 6 个地址位置。这种分配反映了这样的事实，下半部分矢量需要存储 23 个宏块时间，上半部分矢量只需存储 3 个宏块时间。图 7 中所用的术语如下：B0 是指当前正在被处理的宏块，是存储当前正在被解码的宏块的下运动矢量（块 3 和块 4）的地方。B-1 存储上一个被处理的宏块的下运动矢量。同样地，T0 存储当前宏块的上运动矢量（块 1 和块 2），T-1 存储上一个被处理的宏块的上运动矢量。需要特别注意的是这是一种移动的/相对寻址方式而不是绝对寻址。B0 和 B-1 宏块不是象在绝对地址方式中那样编码在绝对地址 0 和绝对地址 1 上。还应注意基地址总是指向 B0（当前正在处理的宏块的下运动矢量）的左运动矢量。这与在图 3 中将基地址计数器 54 中的值乘以 2 所得的基地址是完全相同的。需要特别注意的是，随着计数器的增加，当要开始处理一个新的宏块时，上运动矢量 T0 将成为上运动矢量 T-1。新的上运动矢量 T0 将覆盖下运动矢量 B-23。

同样地，新的下运动矢量 B0 将覆盖旧的上运动矢量 T-2。这样，下运动矢量总是覆盖上运动矢量，上运动矢量总是覆盖下运动矢量。还需注意沿两条不同的时间线为上和下运动矢量分配内存。

下面为上述的概念提供一个例子。例子说明如果计数器的当前值为 5，那么 SRAM 数组 52 中的基地址 10 就是存储当前宏块的下运动矢量的地址。当前宏块的上运动矢量将总是被存储在比计数器多 6 的地址。在本例中即为地址 16。通过仔细观察图 7 的内存分配图我们可以找到原因。因为当前计数总是指向 B0，T0 的相对地址空间通过模加 6 返回 T0 来得到。当处理下一个宏块而计数器增长为 6 时，每个宏块都“长了一个宏块的时间”。基地址 10 将存储上一个宏块的下运动矢量，而基地址 12 存储新的宏块的下运动矢量。需要特别注意的是基地址 12 原来存储的是宏块 C-2（宏块 C-2 包含上运动矢量 T-2 和下运动矢量 B-2）的上半部分矢量。T-2 运动矢量只需存储三个宏块的时间，此时计数器增为 6，B0 的下运动矢量 B0 写到它的位置上。

例子：

		计数器值 5 = 基地址 10						
基地址	0...4	6	8	10	12	14	16	18...
52		B-2	B-1	B0	T-2	T-1	T0	B-23
		计数器值 6 = 基地址 12						
基地址	0...4	6	8	10	12	14	16	18...
52		B-3	B-2	B-1	B0	T-2	T-1	T0

总而言之，整个运动矢量存储和检索模块 38 的操作可以用下面的例子来说明：如果基地址计数器 54 的值为 6，那么它指向的实际地址为 12 因为每个基地址存储两个运动矢量。因此，实际地址 12 存储 B0 的左运动矢量。假设查找表 58 的输入如下：块号 = 1，OBMC/预测模式 = OBMC。那么，块 1 的左，上，右和下 OBMC 运动矢量的检索如下：

MV 选择	位置	LUT 输出	基地址	最终地址
左	T-2	+3	12	15
上	B-23	+8	12	20
右	T-1	+5	12	17
下	B-1	-2	12	10

在上例中,从图6中可以看到块1的OBMC左运动矢量是宏块C-2的右上块,其内存位置为T-2。根据图7或上例可以计算偏移值。在本例中,通过模加3可以看到T-2的右运动矢量。同样地,从图6中可以看到右运动矢量位于内存段T-1的右边,上运动矢量位于内存段B-23的左边,下运动矢量位于内存段B-1的左边。

既然在已经说明了本发明编码操作的装置和方法,我们将说明在解码操作中使用本发明。图1的解码器部分包括视频D-A转换器和编码器26,后处理器24和视频解码器22。

图8是视频解码器22的详图。视频位流作为输入传送给位流解码器70。位流解码器70的第一个输出端连接到用于解码的运动矢量存储和检索模块72的一个输入端。位流解码器70的第二个输出端连接到图2所示的离散余弦变换和反DCT44的一个输入端。用于解码的运动矢量存储和检索模块72的一个输出端连接到图2所示的运动补偿电路42。运动补偿电路42的一个输出端连接到组合电路加法器76的第一个输入端。离散余弦变换和反DCT44的一个输出端连接到加法器76的第二个输入端。加法器76的一个输出端提供4:2:0视频格式的信号。后处理器24的一个输出端连接到视频D/A和编码器26的一个输入端上。后处理器24提供4:2:2视频格式的信号。视频D/A和编码器26提供与图1所示信号相同的视频模拟信号。

在操作中,一个数字视频信号被输入位流解码器70,该解码器从视频位流中分离出运动矢量差和象素数据。运动矢量差被提供给解码用的运动矢量存储和检索模块电路72,该电路将在图9中进一步说明。电路72的输出运动矢量送给运动补偿电路42。在实施完运动补偿之后,得到的象素由加法器76顺序地与离散余弦变换和反DCT电路44的输出相加。位流解码器70提供的象素数据由离散余弦变换和反DCT44处理以进行基于解压缩的变换。离散余弦变换和反DCT电路44的一个输出端作为加法器76的第二个输入端。加法器76提供4:2:0视频格式的信号给后处理器24,该后处理器在从数字到模拟变换之前进行后处理。为得到视频模拟输出信号,进行D/A变换。

图9所示为依照本发明图8中的解码用运动矢量存储和检索模块电

路 72。除了为了说明硬件的实现在复制的相同的电路图上加了一个撇号外，图 9 和图 3 中相同的元素使用相同的参考编号。也就是说，如果编号相同的元素没有撇号，例如运动补偿单元 42，那么该元素由图 3 和图 9 分时共享。图 3 和图 9 非常相似，主要的不同点在于增加了一个加法器电路 80，该电路的第一个输入端用于接收运动矢量寄存器 50' 的一个输出，第二个输入端用于接收预测值计算器 62' 的一个输出。加法器电路 80 的一个输出端提供一个完整的运动矢量（预测值与差的和）用于在运动矢量的 SRAM 数组 52' 中存储。解码用运动矢量存储和检索模块电路 72 的操作与前面讲述的图 2 中的运动矢量存储和检索模块 38 的操作类似。因此，不再重复说明对解码用运动矢量存储和检索模块电路 72 的操作。需要注意的一个类似功能的例外是运动矢量差寄存器 50' 在解码模式下存储一个运动矢量不同于图 3 中的寄存器 50 存储的是一个完整的运动矢量。此外，如图 3 中运动矢量必须存储于 SRAM 运动矢量数组 52' 中。为了得到将在 SRAM 数组 52' 中存储的运动矢量，来自预测值计算器 62' 的预测值必须与来自运动矢量寄存器 50' 的运动矢量差相加。这些运动矢量的存储形成了对应于第二个宏块的第二组运动矢量。

至此很明显本发明提供了一个可以高效地检索和存储运动矢量的改善了的视频运动系统和方法。在内存中存储运动矢量的方法包括接收四个运动矢量的步骤。其中每个运动矢量对应于代表一个视频信号一部分的第一个宏块。第一个和第二个运动矢量对应于第一个宏块的上半部分，第三个和第四个运动矢量对应于该宏块的下半部分。第一个和第二个运动矢量存储在内存的第一部分，第三和第四个运动矢量存储在内存的第二部分。本发明大大地减少了实现 SRAM 运动矢量数组 52 和 52' 所要求的内存量。例如，假设每个运动矢量 2 个字节，视频编解码器的解码器部分的存储需要 192 字节内存。192 的算出方法如下：

$$[(24 \text{ 宏块}) (4 \text{ 运动矢量/宏块}) (2 \text{ 字节/运动矢量})] = 192$$

尽管看来 192 字节内存是所需的最小的内存大小，本发明大大地节省了内存。尤其是，本发明避免存储图 4 所示的宏块 C-2 之前不需要存储的那些宏块的上半部分的运动矢量。也就是说，宏块 C-3，C-4，C-5 到 C-23 的上半部分的两个块都未被存储。这种节省空间的方法要

求的内存大小如下:

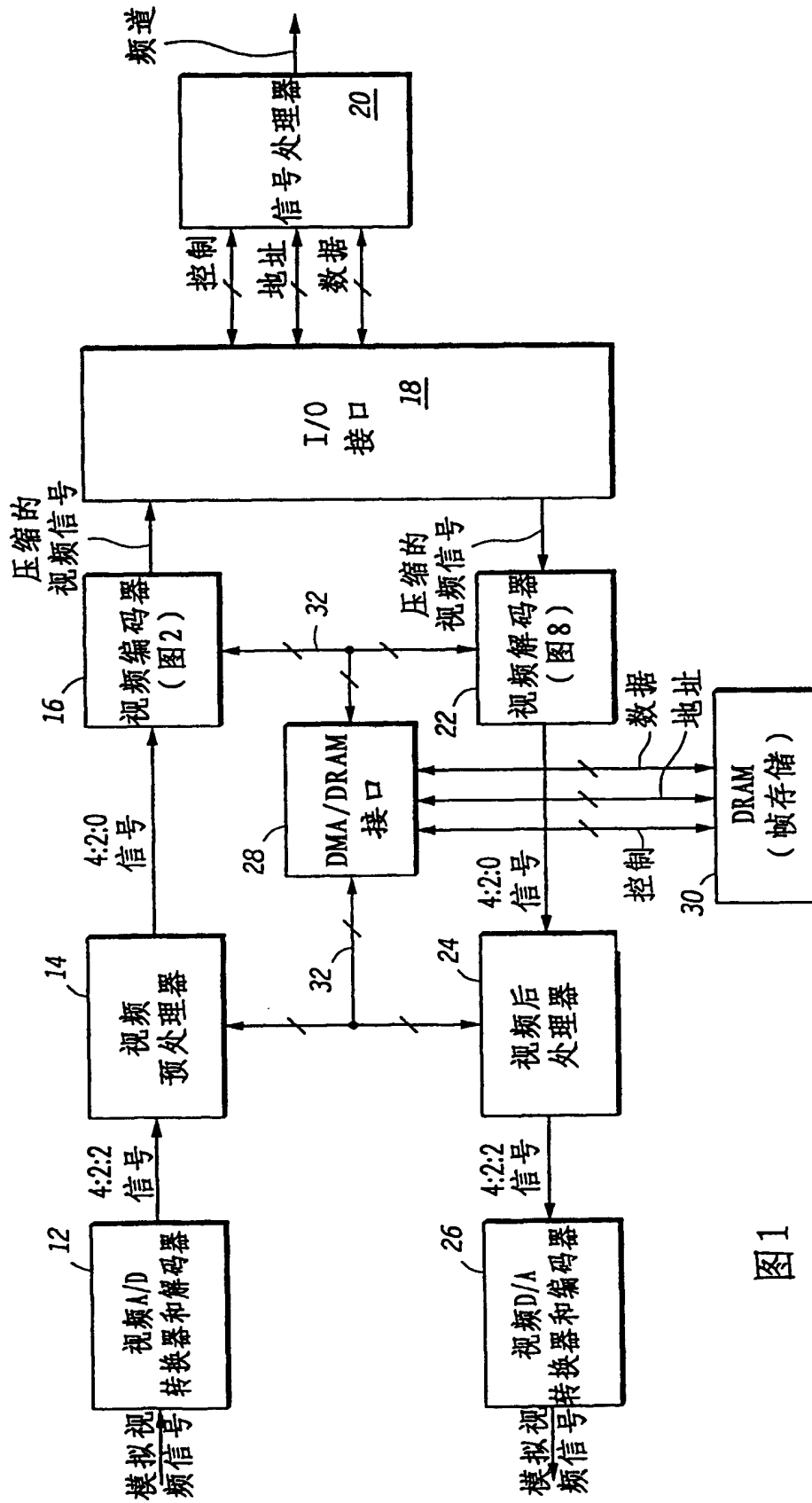
$[(24 \text{ 宏块的下运动矢量})(2 \text{ 运动矢量/宏块})] = 48 \text{ 个下运动矢量}.$

$(3 \text{ 宏块的上运动矢量})(2 \text{ 运动矢量/宏块}) = 6 \text{ 个上运动矢量}.$

$\text{总字节数} = [(48 + 6) \text{ 运动矢量}][2 \text{ 字节每运动矢量}] = 108 \text{ 字节}.$

因为预测器和 OBMC 功能必须在视频编码解码器的编码器和解码器两侧进行, 所以总节省量为 2 (108) 对 2 (192). 在本具体例子中, 正如上面计算的那样, 使用本发明节省内存的大小为对 216 字节对 384 字节 (大约节省 44%). 如果本发明与 HDTV 的 MPEG-4 一起使用, 节省的内存量将会更大, 因为有更高的水平分解率. 需要注意的是使用本文所讲述的内存分配方法丢弃宏块 C-2 之前的块 1 和块 2 并不要求复杂的寻址方法. 本发明允许 H.263 标准的 OBMC 和预测器功能无需增加寻址硬件即可共享内存. 通过减少所需硬件, 硅面积, 成本和电能消耗也能得以减少, 从而使视频编码解码器的硬件实现可与软件实现相竞争甚至优于软件实现.

熟悉本技术领域的人将会认识到可以在不超出本发明的范围内进行修改和变化. 例如, 可以用各种不同类型的电子内存来实现所谓的术语“术存”. 本发明的方法可由各种逻辑门配置来实现. 因此, 本发明可在包含所有这些在所附的权利要求书范围内的变化和修改.



一四

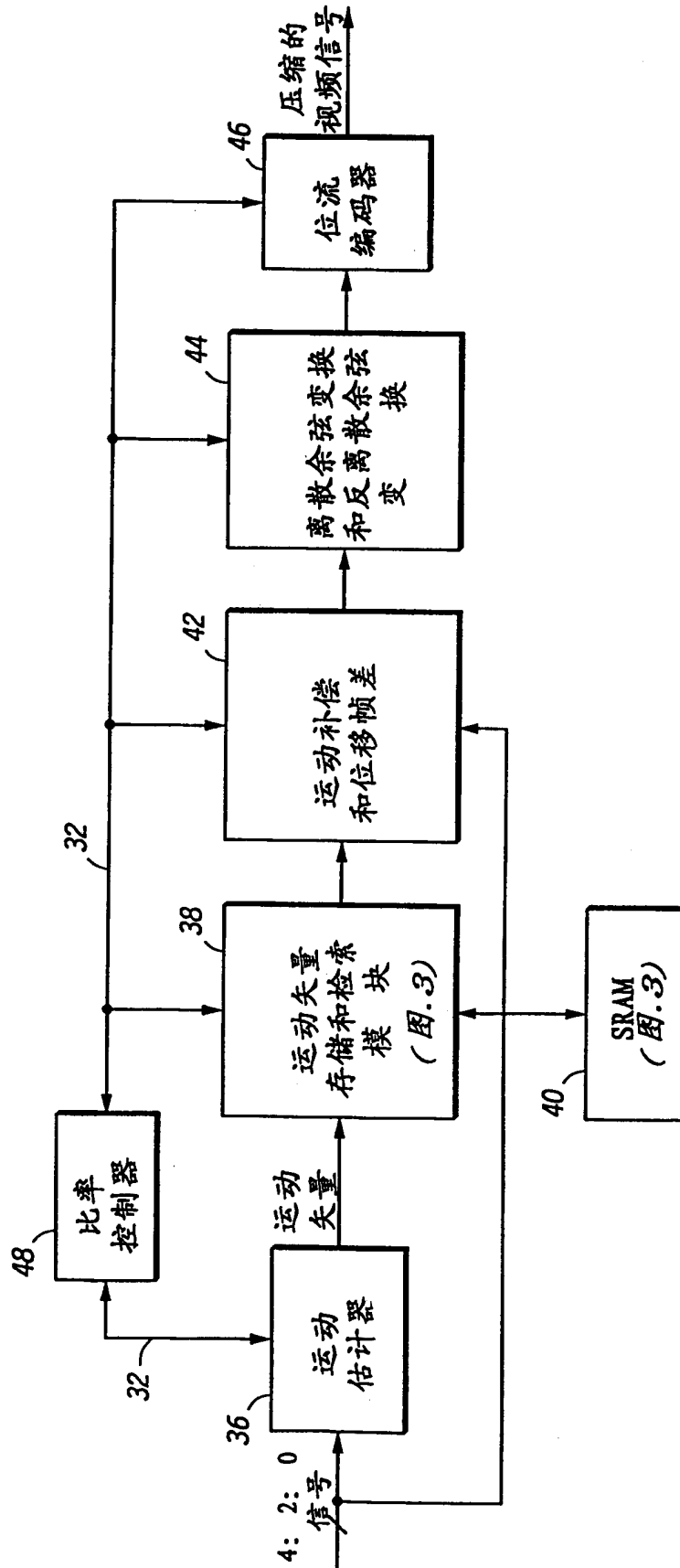
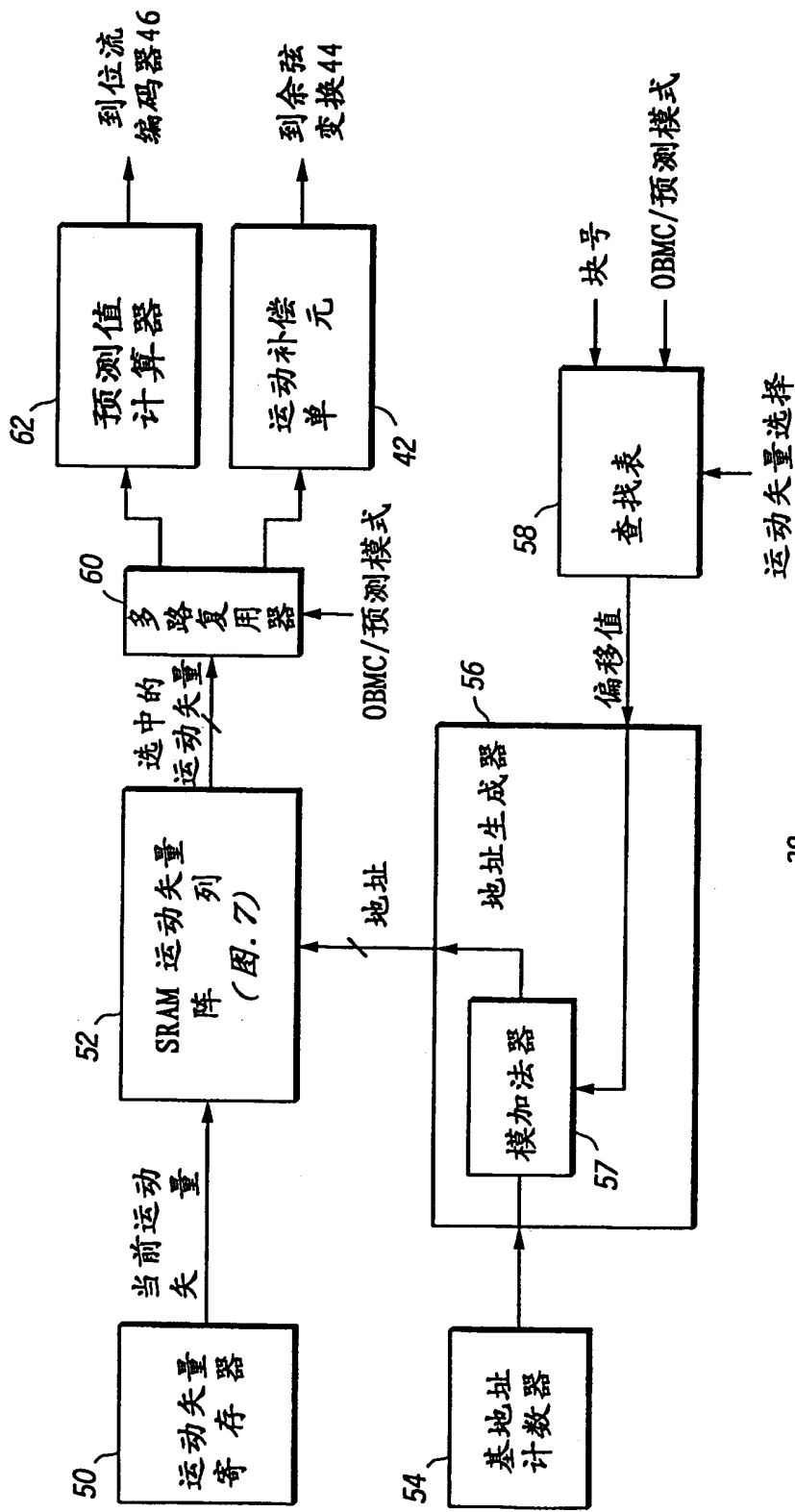


图.2

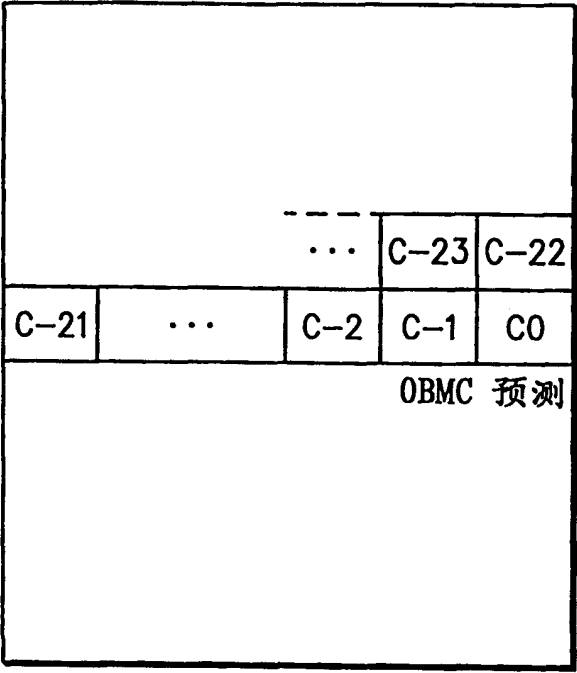
16



38

图.3

图.4



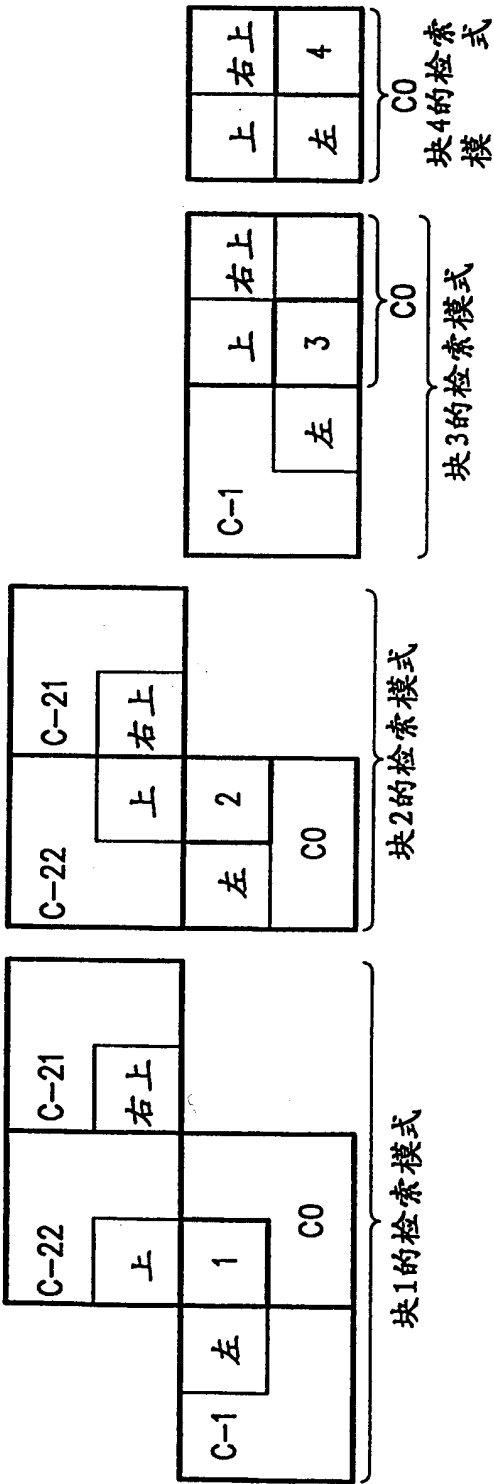


图.5

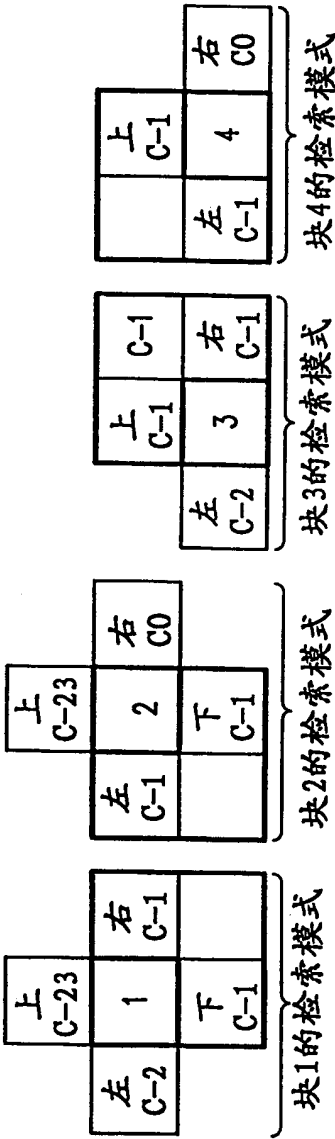
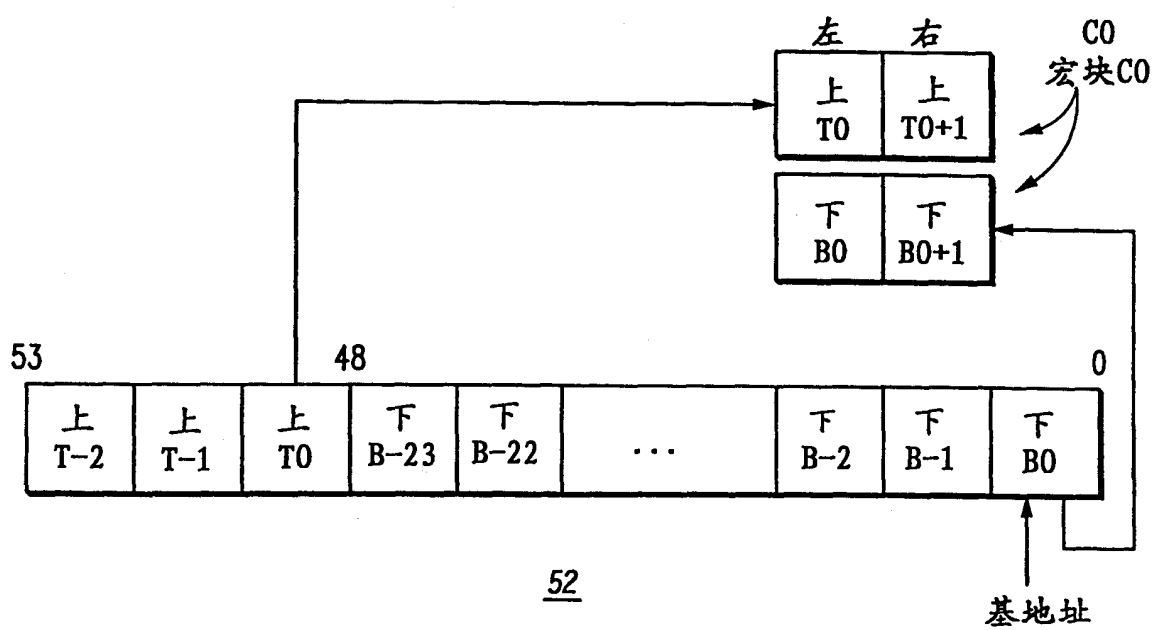


图.6

图.7



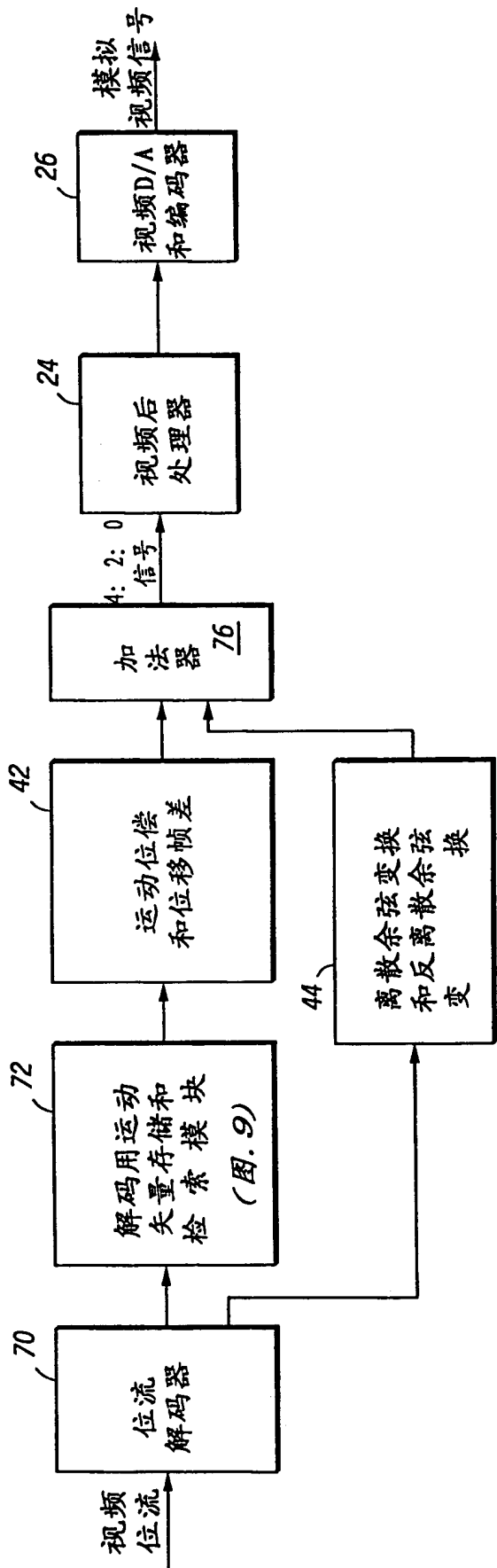


图.8

22

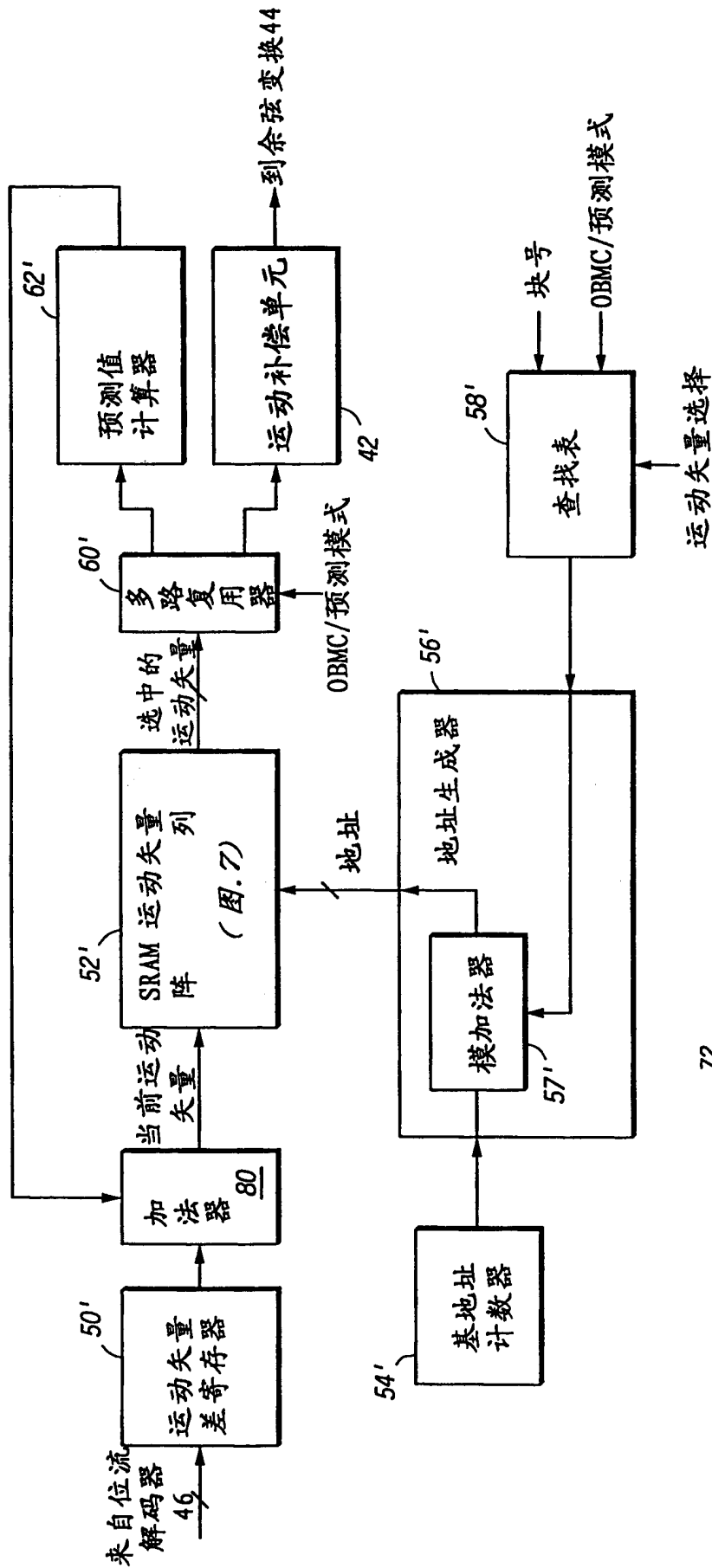


图.9

72