



[12]发明专利申请公开说明书

[21]申请号 92104921.8

[51] Int.Cl⁵

H04N 7/01

[43]公开日 1993年2月10日

[22]申请日 92.6.22

[30]优先权

[32]91.7.22 [33]US [31]733,766

[71]申请人 国际商业机器公司

地址 美国纽约

[72]发明人 宋民少 莱昂·卢梅尔斯基
阿兰·威斯利·皮沃斯
约翰·路易丝·彼得斯[74]专利代理机构 中国国际贸易促进委员会专利
代理部

代理人 付 廉

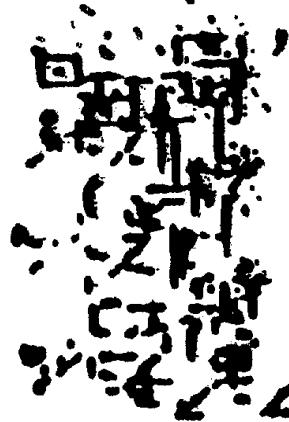
G09G 5/14

说明书页数: 19 附图页数: 5

[54]发明名称 多源图像实时混合和防混淆的方法和
装置

[57]摘要

采用光栅图形显示视频数据通路的方法和装置，实现了多幅图像的任意混合。该视频数据通路并行，并采用在一组查寻表控制下运行的并行装置，查寻表可以从控制器例如主机工作站取出，该通路起提高屏幕分辨率的作用，也能从可能存在的不同图像源中形成各种不同的像素数据格式。在主机工作站的控制下对许多图像源的输出进行混合。转换成共同的预定格式的像素数据格式在由多个图像源产生的像素值之间提供相关性，每个图像源都有一个单独的帧缓冲器。



<45>

权 利 要 求 书

1. 图像显示装置,其特征在于包括:

以分级方式排列的多个帧缓冲器,从而构成一个优先级最高的帧缓冲器和多个优先级较低的帧缓冲器,每个所说的帧缓冲器装置存储从相关联的图像源得到的图像象素数据;

与每个所说的帧缓冲器的输出相连的装置,用于将存储的图像象素数据格式转换成预定的格式;以及

输入端与所说的转换装置的各个输出端相连的装置,用于有选择地将从所说的多个帧缓冲器装置中的每一个得到的图像象素组合,以便形成一个组合图像象素。

2. 如权利要求 1 的图像显示装置,其特征在于所说的组合装置包括多个装置,每一个都放在以分级方式排列的相邻的帧缓冲器装置之间,用于有选择地产生一个图像象素值,以及将产生的图像象素值提供给下一个最高级的发生装置的输入端,所说的发生装置对一个控制装置作出响应,用于:(a)提供一个等于由第一帧缓冲器装置输出的图像象素值的图像象素值,第一帧缓冲器装置与其输入端相连,(b)提供一个等于由第二帧缓冲器装置输出的图像象素值的图像象素值,第二帧缓冲器装置与其输入端相连,(c)提供

一个等于由第一和第二帧缓冲器装置输出的图像象素值组合的图像象素值。

3. 如权利要求 1 的图像显示装置, 其特征在于所说的组合装置包括用于每个帧缓冲器装置的混合装置, 以便根据由相关联的帧缓冲器装置提供的象素透明度值, 对从相关联的帧缓冲器装置得到的象素值和从与下一个较低优先级的帧缓冲器装置相关联的混合装置得到的象素值进行混合。

4. 如权利要求 1 的图像显示装置, 其特征在于每个所说的转换装置都有与控制装置相连的一个输入端, 用于在其中存储代表预定格式的数据。

5. 如权利要求 1 的图像显示装置, 其特征在于对应于 N 个优先级有 N 个帧缓冲器装置, 并且其中 所说的 N 个帧缓冲器装置中的每一个都以 M 路方式交错存取。

6. 一种产生视频象素数据的方法, 其特征在于包括以下步骤:

提供以分级方式排列的多个帧缓冲器, 从而构成一个优先级最高的帧缓冲器和多个优先级较低的帧缓冲器, 每个所说的帧缓冲器装置存储从相关的图像源得到的图像象素数据;

有选择地将从所说的多个帧缓冲器装置中的每一个得到的图像象素组合, 以便形成一个组合图像象素。

7. 如权利要求 6 的一种方法, 其特征在于有选择地组合的步

骤,对每一个存储在帧缓冲器装置中的图像象素来说,是在逐个象素的基础上完成的。

8. 如权利要求 6 的一种方法,其特征在于有选择地组合的步骤包括对预定的象素彩色值连同显示窗口标识进行解码的步骤,以便产生多个控制信号,控制对图像象素有选择地组合的过程。

9. 如权利要求 6 的一种方法,其特征在于至少一个帧缓冲器装置以与已经存储在其中的每个象素值相结合的方法,存储一个透明度值,其中有选择地组合的步骤将透明度值、相关联的象素值和来自下一个较低优先级帧缓冲器装置的经组合的象素值组合在一起。

10. 如权利要求 6 的一种方法,其特征在于提供分级方式排列的多个帧缓冲器的步骤包括将从每个帧缓冲器装置得到的存储的图像象素数据的格式转换成预定格式的步骤。

11. 一种用于有选择地将从第一输入端和第二输入端得到的图像象素数据混合在一起的集成电路器件,其特征在于所说的集成电路器件包括第一装置、第二装置和第三装置,第一装置将从第二输入端得到的象素值中减去从第一输入端得到的象素值,第二装置将所说第一装置输出的差值乘以与从第一输入端得到的象素值相关联的象素透明度值,第三装置将所说第二装置输出的乘积与从第二输入端得到的象素值相加。

12. 如权利要求 11 的一种集成电路器件,其特征在于延时装

置,用于对从第二输入端得到的象素值同步延迟一个第一、第二装置进行操作所需要的时间量。

13. 图像显示装置,其特征在于包括:

以分级方式排列的多个帧缓冲器,从而构成一个优先级最高的帧缓冲器和多个优先级较低的帧缓冲器,每个所说的帧缓冲器装置存储从相关联的图像源得到的图像象素数据;

放在以分级方式排列的相邻的帧缓冲器装置之间的装置,用于产生一个图像象素值,以及将产生的图像象素值提供给下一个最高级的另一发生装置的输入端,所说的发生装置对控制信号作出响应,用于(a)提供一个等于由第一帧缓冲器装置输出的图像象素值的图像象素值,第一帧缓冲器装置与其输入端相连,(b)提供一个等于由第二帧缓冲器装置输出的图像象素值的图像象素值,第二帧缓冲器装置与其输入端相连,(c)提供一个等于由第一和第二帧缓冲器装置输出的图像象素值组合的图像象素值。

14. 如权利要求 13 的图像显示装置,其特征在于与每个所说的帧缓冲器装置的输出端相连的装置,用于将存储的图像象素数据格式转换成一个预定格式。

15. 如权利要求 13 的图像显示装置,其特征在于所说的发生装置对一个相关联的的说的帧缓冲器装置输出的透明度值作出响应,以便根据该象素透明度值将第一和第二帧缓冲器装置输出的图像象素值进行混合。

16. 如权利要求 13 的图像显示装置，其特征在于所说的控制信号由包括查寻表装置的控制装置产生，用于根据象素彩色码和显示装置窗口标识，以逐个象素为基础产生控制信号，其中所说的象素彩色码和显示屏幕窗口标识存储在与所说控制装置相关联的一个所说的帧缓冲器装置中。

17. 如权利要求 13 的图像显示装置，其特征在于所说的发生装置包括第一装置、第二装置和第三装置，第一装置将从与处于下一个较低优先级的帧缓冲器装置相关联的发生装置得到的象素值中减去从相关联的帧缓冲器装置输出的象素值，第二装置将所说第一装置输出的差值乘以与从相关联的帧缓冲器装置输出的象素值相关联的象素透明度值，第三装置将所说第二装置输出的乘积加上从与处于下一个较低优先级的帧缓冲器装置相关联的发生装置得到的象素值。

说 明 书

多源图像实时混合和防混淆的方法和装置

一般地说，本发明涉及图像显示装置和方法，特别是涉及对来自多个图像源的图像同时显示。

在一个新式显示系统中，可以显示来自多图像源的数据。这些图像是被利用所谓“窗口”来进行显示的，在这种窗口技术中，一个图像源分配给一个显示区。然而，当多图像源的输出必须协调显示时，问题就出现了。当有一个以上的图像源需要显示时，需要一种机制来协调单个显示监视器上的输出。

一种直截了当的解决办法是设计这样的系统，使其在某一时刻只能显示一个图像源。例如，如果选择了高清晰度电视(*HDTV*)的取样器输入，那么只有 *HDTV* 的输出送至显示器。然而，随着在图形系统中对于“窗口”需要的增加以及对于存在的多于一个视频源的多媒体环境的重视，这种相对简单的解决办法就不行了。

为了实现窗口，必须有这样一种机制，以便在由一个监视器屏幕确定的全部区域中，将不同的子区域分配给不同的视频源。图 1 表示一个开窗口的图形监视器屏幕的例子。区域 A 可以分配给本机主机

或工作站。区域 B 可以通过一个高速网络分配给与之相连的远程图形服务器。区域 C 可以分配给 HDTV 的取样器。在本领域中提供这种显示的技术称作象素切换。即对一给定的显示区即窗口而言，该区域的象素源是从特定的图像平面中选取的。

当从图像源中选出的任意形状的图像叠加在另一图像的上面，而不是显示在一个简单的矩形窗口的上面时，就会出现更复杂的问题。例如，利用图形服务器可以提供汽车的图像，该汽车图像可能需要叠加在一幅由 HDTV 产生的背景图像上。由于前景物体即该汽车的形状不是长方形的，所以全部象素的选择就需要在逐个象素的基础上完成。

适用于两个图像源系统的解决办法是采用彩色键控象素切换，允许在两个图像源之间进行逐个象素的选择。这种技术在 L. Lumelsky 等人的被授予的美国专利 No. 4,994,912“音频视频交互显示”中有过说明。

然而，如果出现如图 1 所示的视频源大于两个的情况时，为了逐个象素的基础上显示 $N(N > 2)$ 个图像就需要一种不同的解决办法。

重叠任意形状的多源图像所带来的另一个问题是起因于混淆效应，这是通过图像象素切换产生的。在这种情况下，一个图像源的一个象素可能不与第二个图像源的一个象素融合，于是便导致混淆。混淆使得最终得到的图像沿前景和背景之间的边界呈现所不希望有的效果，比如阶梯形边界或彩色失真。为了消除这种效果，就需要一

种防混清技术。然而，在前景图像不包括背景图像的信息的情形中，防混清应当在视频输出中以实时方式完成。于是，就需要一种对 N 个图像源进行实时防混清的技术。此外，为了对一个任意形状的前景物体进行防混清，必须在逐个象素的基础上完成防混清。

然而，在现代高分辨率显示中，视频数据的带宽很宽，数据速率很高，因此对于必须以实时方式进行任何象素的处理均实行严格实时限制。

在 *Pappas* 等人的于 1991 年 3 月 19 日公布的美国专利 No. 5, 001, 469“窗口从属缓冲器选择”中，描述了图形子系统中的窗口控制硬件，其中多图像源显示在单个监视器上。处理方式如下：为每个窗口确定一个单独的帧缓冲器，为每个窗口即帧缓冲器确定窗口标识、窗口尺寸和基于上、下、左、右位置四个量的窗口位置。该系统还采用按优先序排列的方案，其中 N 个帧缓冲器（“窗口”）从 0 到 $N-1$ 按优先序排列，0 优先级最高， $N-1$ 优先级最低，该图形子系统包括 N 个窗口检测逻辑电路，每个帧缓冲器一个，该逻辑电路利用比较窗口尺寸和位置量的比较器来确定相关的窗口是否是屏幕区域中的有效区域。如果该窗口有效，那么一个图像源指针和其它信息被送入对 N 个输入信号按优先序排列的逻辑电路，以确定哪个有效的图像源优先级最高。由优先级逻辑电路选出的具有最高优先级的有效图像并显示在监视器上。

Pappas 等人利用窗口尺寸和位置量控制多图像源，并且一个图

像帧缓冲器不包括多个窗口。此外，该系统似乎只限于采用矩形窗口，显示具有不同图像格式的多图像源带来的问题也没有论及。

因此本发明的一个目的是在对于 N 个图像源的逐个像素的基础上，通过采用象素切换和控制的方法，来实现来自 N 个 (N 可以大于 2) 独立图像源的视频数据的同时显示。

本发明的另一个目的是在逐个象素的基础上，根据象素彩色键控和窗口标识，利用 α 混合和象素切换相结合的方法，提供一种在一个监视器上显示多个图像源的视频数据的方法和装置。

通过采用光栅图形显示视频数据通路的方法和装置，能够解决上述其它问题，从而实现本发明的目的，该通路实现了多幅图像的任意混合。视频数据通路高度并行，采用在一组查寻表控制下运行的并行装置。查寻表可从控制器例如主机工作站取出。光栅图形显示视频数据通路的工作不受屏幕分辨率的限制，也能从可能存在的大量的不同图像源中形成各种不同的象素数据格式。在主机工作站的控制下对许多图像源的输出进行混合，象素精度则基于：(a) 每个图像源图像的组合透明系数(α)以及(b)由主机工作站分配的窗口标识数。

转换成共同的预定格式的象素数据格式在由多个不同图像源产生的象素值之间提供相关性，这些图像如 HDTV，以及超级计算机那样的高分辨率图形服务器。

防混淆功能用来消除前景图像和背景图像交界处的非自然效

果,防混淆功能还能用来产生特殊效果,比如在来自一个图像源的背景图像的上部显示来自另一个图像源的半透明图像前像。实现防混淆的最佳办法是利用象素混合。

为了允许任何数目的独立的视频源存在,对每个视频源就要指定一个单独的帧缓冲器。这样,有 N 个独立的视频数据源的系统,就有 N 个帧缓冲器。依照显示监视器的带宽要求, N 个帧缓冲器中的每一个也可以以 M 路方式交错存取。

本发明还与一个集成电路器件有关,它以高速流水线方式实现象素混合功能和象素多路传输功能。

通过结合附图对本发明所作的详尽描述,本发明的上述特征及其它特征将变得更明了,其中:

图 1 表示具有多个视频数据源,每个又在相应的窗口中显示的显示屏幕的实例;

图 2 是本发明的多个分级排列的帧缓冲器的框图,这些帧缓冲器有查寻表,以及放在帧缓冲器之间的象素混合和象素多路传输电路;

图 3 是表示结构为 $N \times M$ 的分级排列的帧缓冲器的框图,这些帧缓冲器有查寻表,以及放在帧缓冲器之间的象素混合和象素多路传输电路;

图 4 表示本机主机视频通路;

图 5 是表示装在一个集成电路器件中的象素混合和多路传输逻

辑电路的框图；以及

图 6 表示从图 5 所示象素混合逻辑电路得到的 α 控制逻辑功能。

图 2 表示控制本发明构造和运行的多源视频显示象素混合系统 10 的框图。系统 10 从 N 个视频源接收视频数据输入信号，每个视频源都有一个相应的帧缓冲器(FB)存储器(FB—1 至 FB—N)。图中还画出了每个 FB 的一个象素(x, y)的视频输出数据通路。由于有 N 个重复的 FB，所以对一个在显示器 12 处的位置(x, y)显示的象素来说，就有 N 个重复的象素，每个象素都来自一个 FB。这些 FB 按从 1 到 N 的次序分级排列，其中 1 被认为是最高优先级，而 N 被认为最低优先级。最好(但不作为对本发明实际应用的限制)FB—1 与主机 14 相连，FB—N 和例如 HDTV 视频源相连，该视频源不提供象素的半透明度(即 α)的标识。FB 以并行编址，因此它们的操作具有彼此相固定的关系。

在如图 2 所示的一种多媒系统中，并不能假定象素格式的相关性。例如，由超级计算机和/或图形服务器产生并且从通讯网络接收的图像可能具有 24 位红、绿、蓝(R、G、B)格式。然而，本机工作站主机 14 可能具有 8 位象素格式，而一个 HDTV 取样输入可能具有色度/亮度(Y/C)格式。为了对从这些不同的视频源 得到的象素混合，第一步就要把象素值转变成一个共同的格式。在系统 10 中，虽然其它格式也在本发明的范围之内，还是选择 R、G、B 格式作为共同的格

式,因为它相对比较简单。这样,所有的象素数据被转换成 R、G、B。数据转换通过使用查寻表(LUT)15,在每个 FB 完成。这就是说,由于已知存储在给定 FB 中 的特定格式,因而对相连的 LUT15 按此方式编程;即,当着要被转换格式的象素加到 LUT15 的地址输入端时,则输出相应的 R、G、B 值。根据具体应用,虽然也可以采用一些固定内容的 LUT,但是每个 LUT15 最好与本机主机 14 相连,因而是可编程的。应该懂得,如同在图 1 中窗口 A 显示的数据那样。主机 14 也可以是一个 FB 的一个视频数据源。作为一个例子,如果主机 14 是一个工作站,那么用于显示的字母数字文本信息可以由主机 14 产生。

还应该注意到,对某些视频源来说,例如 HDTV 视频源,视频信号可以用一个模拟格式来表示。因此,要用一个模/数转换器先将 模拟视频输入信号转换成适合于在 FB—N 中存储的数字格式。

因为在象素混合之前就形成了共同的象素格式,采用:LUT15 象素数据格式转换器解决了由每个独立的图像源提供的象素数据格式的相关性问题。

为了对每个象素边界进行混合和防混淆处理,除了 FB—N(它不具有下方 FB,因此,没有背景图像与之相混合),每个 FB 都包括一个 α 缓冲器存储器级。 α 缓冲器存储器级由与 FB 相连的也是视频数据源的装置存取。例如,FB—2 可以有 32 位存储器级,由 24 位的 R、G、B 级和 8 位的 α 级构成,二者都由图形服务器在整个通讯网络

中追寻。当图形服务器提供一幅图像时，它也产生一个与每个被提供图像的象素有关的 α 值。这里用 α 表示象素的半透明度或透明度，并假定取值在 0 和 1 之间。如果 α 等于 0，则相关的象素被认为是完全透明的。这就是说，任何背景象素都是可见的。如果 α 等于 1，则相关的象素被认为是完全不透明的。0 和 1 之间的值用来表示象素透明度的逐渐变化情况，采用这些值以改善在前景和背景图象的交界处的状况，以防止混淆边界象素，从而消除所显示的不希望出现的非自然效果。

象素数据和 α 值都经通讯网络传递至相应的 FB—2。8 位 α 值通过 MIX 逻辑电路 16 和多路传输(MUX)逻辑电路 17 用来进行象素混合，结果，从下方 FB 向上传过来的 R、G、B 图像象素数据与 FB—2 得到的象素混合。 α FB 级从而能对前景象素和背景象素进行实时的、逐个象素的混合。防混淆也能以实时方式完成，这将在以后说明。

在工作过程中，具有最低优先级 (N) 的 FB 通过相关的 LUT15 将象素 (x, y) 转换成 (R, G, B) 象素格式，并向上传送经转换的象素值，通过 MIX16，将其与从相邻的高一级优先级的帧缓冲器 ($FB-N-1$) 中得到的相应的象素混合。除了第 N 个 FB (优先级最低)，所有 FB_i 都能执行三种操作中的一种操作，这些操作如表 1 所示。

表 1

放弃从第 $j+1$ 个 FB 传过来的象素，并向上方的第 $j-1$ 个 FB 传送相应的象素，如果 $j=1$ ，则将相应的象素传给显示器 12。这一操作是通过选择相应的 $MUX17$ 的 A 输入完成的。

2. 放弃本身的象素，并向第 $j-1$ 个 FB 传递从第 $j+1$ 个 FB 得到的象素，如果 $j=1$ ，则将相应的象素传给显示器 12。这一操作是通过选择相应的 $MUX17$ 的 C 输入完成的。

3. 采用从相应的 FB 得到的 α 值，将本身的象素值与从第 $j+1$ 个 FB 中得到的象素混合，然后将混合后的象素值传送给第 $j-1$ 个 FB ，如果 $j=1$ ，则将相应的象素传给显示器 12。这一操作是通过选择相应的 $MUX17$ 的 B 输入完成的。

上述方法提供了这样一种技术，其中对一个给定的显示象素 (x, y) 来说，或者从任何 FB 选择一个单独的象素 (x, y) （象素切换），或者从不同的 FB 选择两个或两个以上象素 (x, y) 的任何混合结果（象素混合）。

参看图 4，本发明采用基于本机主机 14 的象素值的彩色键控方法。本机主机 14FB 由三級组成。第一級存储本机主机 14 的象素值，即彩色索引(CI)。第二級存储本机主机 14 的窗口标识数(WID)。第三級存储本机主机 14 的 α 值(WS- α)。系统 10 采用控制逻辑电路 19，它根据从主机 14FB 得到的输入，产生多个信号。这些输入包括用于彩色键控”的 CI 输入，以便确定显示器 12 的操作。WID 提供了

这样一种机制,其中显示器 12 的不同窗口可以具有与之相关的一个或多个键控彩色。这一过程通过采用调色 *LUT15* 和视频控制 (VC) *LUT22* 完成。*VC-LUT22* 具有 $2(N-1)$ 个输出,它们成对地提供给每个 *MUX17*,以便根据表 1 控制 *MUX17* 的操作。

例如,对第一窗口来说,红色 *CI* 定义为“键”通过 *VC-LUT22* 选择 *HDTV* 的输入。对另一窗口来说,红色 *CI* 可以是这样的“键”,它通过 *VC-LUT22* 将主机 14 的象素与 *HDTV* 背景象素混合,采用 *WS- α* 修改象素的边界值,以便进行防混淆处理。对再一个窗口来说,红色 *CI* 可以是一个在监视器 12 上显示的象素,通过调色 *LUT20* 将 *CI* 转换成 24 位 *R、G、B* 格式。这一方法有助于任何数目的独立的图像源之间的协调,而不是仅限于两个图像源。此外,该方法也提供了象素混合功能,例如实时防混淆或图像融合。所有这些功能可以根据 *WID* 的值,在包含于感兴趣的物体中的象素上和/或与感兴趣的物体为界的象素上实现。再者,这些功能是在逐个象素的基础上实现的。

视频输出控制最好采用本机主机 14*FB* 完成。为了说明起见,选择 *FB-1* 作为本机主机 *FB*,但是也可以代之以选择任何一个 *FB*。从图 4 可以看出,对本机主机 14 图形工作空间来说,*FB* 总共有 *P* 位级。这些级的 *P* 位输出中,*C* 位用作彩色索引(*CI*),*W* 位用作窗口标识(*WID*),以及 *M* 位用作将本机主机 14 的图像与另一图像融合的(*WS- α*)。*CI* 和 *WID* 二者都用作 *LUT15* 的索引(地址),

以便提供 24 位 R、G、B 数据。此外，这些相同的位用作 VC—LUT22 的索引(地址)，以便提供视频通路控制位(VID—CNTRL)，它们用来协调显示输出。VID—CNTRL 的宽度是图像源(N)总数的函数。

对每个 FB_j 来说。(其中 $1 \leq j < N$)，需要两位来选择表 1 中所述的三种操作中的一种。这两位定义如下：

00——禁止混合，选择 MUX 的输入 A；

01——禁止混合，选择 MUX 的输入 C；以及

1X——允许混合，选择 MUX 的输入 B。

对 FB_j 来说，如果位 1 置“1”(MSB)，则允许混合， FB_j 和 FB_{j+1} 象素混合的结果传送给第 $j-1$ 个 FB。在这种情况下，位 0 (LSB)忽略不计。如果位 1 清零，那么禁止 α 混合，要么是从 FB_j 得到的象素值(如果位 0 清零)，要么是从 FB_{j+1} 得到的象素值(如果位 0 置“1”)被传送至 FB_{j-1} 。

既然 $FB-N$ 仅能将其输出的象素值向上传送，所以它就不需要任何控制位。这样，对 N 个图像源来说，所需要的 VID—CNTRL 的总位数是 $B=z(N-1)$ ，其中 B 是 VID—CNTRL 的位数， N 是独立的图像源数。

对 FB_j 分配的 VID—CNTRL，最低有效位是 z_j-2 ；最高有效位是 z_j-1 ，这为多源系统提供了灵活的窗口控制机制。应该注意的是，通过采用彩色键控，用户从本机主机 14 可以从所有 FB 得到的

相应象素的结果形成任何显示象素。这就是说，用户定义一个彩色键以代表一个特定的输出格式，然后当需要一个特定输出时，利用该彩色键。此外，通过使用 **WID**，用户能够选择作为窗口标识函数的彩色键控。例如，如果 **WID** 的宽度是 4 位，那么最多可以有 2^4 即 16 个窗口同时被显示和控制。为此，**VC-LUT22** 最多包括 16 个不同的存储区，每个存储区包含不用的 **VID-CNTRL** 值。**VC-LUT22** 中的一个特定区域由 **WID** 的值选择，而区域中特定的 **VID-CNTRL** 位的组合由 **CI** 的值选择。

总之，这种机制为任何数目的 **FB** 提供了灵活的输出控制，这些 **FB** 以逐个象素为基础而采用了彩色键控和混合（防混淆）技术。如果每一级都允许进行 **FB** 混合，那么最终显示的象素(**R**)为：

$$R = \alpha_1 P_1 + (1-\alpha_1)(\alpha_2 P_2 + (1-\alpha_2)(\alpha_3 P_3 + (1-\alpha_3) \dots \\ \dots (\alpha_{N-1} P_{N-1} + (1-\alpha_{N-1})P_N) \dots \dots)$$

其中 **Pj** 代表从 **FBj** 得到的象素， α_j 代表从 **FBj** 得到的 α 值。

一个高分辨率监视器（例如 2048×2048 象素）需要超过 $360MHz$ 的视频带宽。为了给一个高分辨率监视器提供所必需的带宽，从 **FBVRAM** 得到的串行视频输出需要交错存取。例如，对一个常规的 **VRAM** 来说，串行输出带宽为 $33MHz$ ，一个 $60Hz$ 的 2048×2048 分辨率的监视器，**FB** 视频输出通路需要至少 11 路交错存取。于是，这就需要有 11 条独立的数据通路。作为交错存取的另一个例子，对一个分辨率为 1280×1024 的监视器来说，视频带宽为

$110MHz$ 。因此,对这种分辨率 4 路交错存取就够了。然而,对一个分辨率为 2048×1536 的监视器来说,视频带宽为 $260MHz$ 。这就需要 8 路交错存取,因为 4 路交错存取仅提供 $4 \times 33MHz$ 即 $132MHz$ 的带宽,而 8 路交错存取提供 $264MHz$ 的带宽。

每个 **FB** 的 **MIX** 逻辑电路 16 和 **MUX** 逻辑电路 17 重复出现,数目与交错存取因数一致。一般地,对于每个交错存取的、独立的数据通路来说,都有一个 **MIX** 逻辑电路 16 和 **MUX** 逻辑电路 17。于是,对 N 个图像源来说,如果 **FB** 的显示输出进行 M 路交错存取,就有 $((M \times N) - N)$ 对 **MIX** 和 **MUX** 逻辑电路块,因为优先级最低的 **FB** 级不需要混合。这种模块方法能够对任何分辨率的监视器进行实时像素混合。

图 3 表示一个交错存取的视频系统的 MXN 实施例。串行处理器 24 接收从 $j=1$ 混合器 30 来的输出信号,并将它们以视频时钟的速率传送至显示器 12。

上述集成电路装置可以通过几种方法实现。一种方法是在对从 **FB VRAM** 得到的交错存取视频数据输出进行串行处理后,提供一个快速集成电路器件。然而,对 $260MHz$ 显示来说,这需要 3.85 毫微秒的周期去进行任何操作,例如融合两幅图像。另一种方法是在每条交错存取存储器数据通路的输出端提供多个较慢的器件,如图 3 所示。第二种方法优于第一种方法,因为第一种方法或需要 **ECL** 门矩阵或需要 **GaAs** 门矩阵以适应宽的视频带宽。然而,以下描述的同

样电路结构将使之适合于任何一种方法。

视频数据通路流中最占时间的部分是进行融合处理,因为这需要乘法器和加法器。 FB_j 是从前一个 FB_{j+1} 的融合功能采用以下等式:

$$R_j = (\alpha_j P_j) + ((1 - \alpha_j) R_{j+1})$$

其中 R_j 是输出至下一个 FB 的所得到的象素值, P_j 是从 FB_j 得到的象素, R_{j+1} 是从前一个 FB_{j+1} 输入的象素值, 以及 α_j 是 $P_j / (P_j + R_{j+1})$ 象素加权, 其中 ($0 \leq \alpha \leq 1.0$)。这种方法需要两个乘法器和加法器。然而, 通过代数运算得到:

$$R_j = (\alpha_j P_j) + ((1 - \alpha_j) R_{j+1});$$

$$R_j = (\alpha_j P_j) + R_{j+1} - (\alpha_j R_{j+1}); \text{ 以及}$$

$$R_j = (\alpha_j (P_j - R_{j+1})) + R_{j+1}$$

末尾的 R_j 表达式只需要一个乘法器和两个加法器。

图 5 表示一个混合器的集成电路器件 30 的框图。由于所要求的时间周期的原因, 加法器 32 和乘法器 34 是流水线的, 并且所有数据值都被延迟, 以便它们保持同步, 这在以下将要说明。

以下描述图 5 中 MIXER30 的功能。同时也参照图 3。与每个象素相关的图像数据同时到达输入端。为了提供最快的时钟速度, 所有输入和输出都被寄存 (REG1 至 REG8), 并被系统时钟计时。

对一个 M 路交错存取存储器来说, 系统时钟频率通过用 M 去除视频带宽得到。例如, 在使用一个分辨率适中的监视器的情况下, 视频带宽为 $110MHz$ 。此例中, 如果 FB 存储器是 1 路交错存取, 则系统时钟为 $110MHz/4$, 即 $27.5MHz$ 。在使用一个高分辨率的监视器的情况下, 视频带宽为 $260MHz$, 并且存储器通过 8 路交错存取。这样, 系统时钟为 $260MHz/8$, 即 $32.5MHz$ 。

要想使全部乘法器都工作在大约 $33MHz$ 的条件下, 且又非常经济, 这是很困难的, 既然如此, 乘法器 34 最好以流水线方法工作。流水线的深度取决于所选择的芯片技术。采用容易获得的集成电路技术, 乘法器流水线深度可以达到两级, 而加法器 34 可以达到一级。为了最大限度地提高运算能力, 每个功能块的输出都被重新同步。这样, $MIXER30$ 与一个单独的自由运行的系统时钟完全同步。

下面说明 $MIXER30$ 的功能。所有输入同时计时, 进入芯片寄存器 $REG1$ 至 $REG4$ 。这些输入是 $NID-CNTROLj$, α_j , R_j 和 R_{j+1} 。 $REG3$ 和 $REG4$ 的输出送入功能块 $FUNC1$, 进行 $(R_3 - R_4)$ 的运算。同时, 为了和所有其它的输入同步, R_1 、 R_2 和 R_4 经寄存器 $REG5$ 和 $REG6$ 延时。 $FUNC4$ 的功能以后讨论。在紧接着的两个时钟周期内, $R5$ 和 $F1$ 通过 $FUNC2$ 相乘, 并且 $R6$ 通过 $REG7$ 和 $REG8$ 相应地延迟两个时钟周期。最后, $F2$ 和 $R8$ 送入功能块 $FUNC3$, 进行 $(F2 + R8)$ 的运算。

为了得到全部 N 个帧缓冲器的最佳混合, 应当提高随着在每个

FB_j 处的 P_j 的宽度的增加而提高的 R_j 的总精度。然而,这不是一个实际的解决办法,因为这不能有效地将全部 R_j 的清晰度带到下一级。不过,可以采用一种简单的舍位法,结果在每一混合级平均产生 $1/2$ 位的误差。

如果在功能块 **FUNC5** 进行舍入操作,那么平均误差是 $1/4$ 位此值是如果不用 **FUNC5** 时的误差 $1/2$ 的一半。这样 **FUNC5** 提供的最终得到的象素 R_j 与 P_j 的位数一样。因此, **MIXER30** 所需的全部时间占六个系统时钟周期。

MIXER30 有两个作用。第一个作用是进行 α 混合(即防混淆)

第二个作用是数据通路控制,如图 2 所示,除了 $FB-N$,每个 FB 都有一个“三对一”的 **MUX17**。为了进一步降低成本和减小复杂程度, **MUX17** 的功能包含在 **MIXER30** 之中,这在以后说明。

当 α 设定为 1.0 , $R_j(\alpha$ 混合的结果)为 P_j 。当 α 设定为 0.0 , R_j 为 R_{j-1} 。换句话说,混合功能也包括了一个乘法器的功能,只要适当选择 α 的值就可以了。如图 5 所示,这是在功能块 **FUNC4** 中完成的。对于一个 8 位宽的 α 值, **FUNC4** 的逻辑电路如图 6 所示。用两位的 **VIDEO-CNTRL** 控制 α 值。当 α 启动位 (**NID-CNTRL-1**) 为高电位,则 α 输出与 α 输入相同。当 α 启动位 (**YID-CNTRL-1**) 为低电位, $\alpha-OUT-6$ 至 $\alpha-OUT-0$ 则被与门 **AND6** 至 **AND0** 强置为“0”。此外, $\alpha-OUT-7$ 设定为“1”,使 $\alpha-OUT=1.0$,这使 **MIXER30** 选择 R_{j+1} 。

假定一个 8 位 α 值, MIXER30 的输入和输出总数目为:

对每一种 $R、G、B$ 颜色: $Rj8$ 个输入;

$Pj8$ 个输入;

$R_{j+1}8$ 个输入;

共同的信号: $\alpha8$ 个输入;

VID—CNTRL2 个输入;

系统时钟, 1 个输入,

总共有 83 根引线, 但不包括电源等。

如前所述, 进行象素混合所遇到的一个问题是由于两个不同的视频源产生的两幅图像的格式不一样。例如, 两个图像源之间的象素形状可能不一样, 其中一个图像源的象素可能是方的, 而另一个不是方的。另一个例子是两种图像源的分辨率可能不一样。因此, 为了对全部的 FB 来说是逐个象素在大小、形状上都逐一对应, 如果一个图像源小于另一个, 则它的图像需要进行放大; 如果一个图像源大于另一个, 则它的图像需要进行缩小, 根据本发明的精神, 解决这些问题的一个办法是对所有的 FB 进行超取样, 其中 FB 的图像源的分辨率是每个图像源的分辨率的最小公倍数, 不论是在水平方向还是在垂直方向上。最小公倍数 FB 方法是最佳的, 因为在使用 $HDTV$ 图像取样器的情况下, 分辨率固定在 1920×1035 象素, 超取样不总是容易实现的。然而, 如果 FB 的大小等于最小公倍数, 那么作为超取样的方法, 象素可以很容易地复制或插入。为了进行超取样, 另一种

方法是采用数字信号。

然而应该注意的是，最小公倍数方法可能导致非常大的 FB 。例如，480 扫描行和 1035 扫描行的最小公倍数是 33,120 扫描行。此外，最终得到的象素可能不是方形象素。还需要进行时基校正来与监视器带宽相适应。

值得庆幸的是，许多图形系统或视频系统的象素通常是方的（或几乎是方的）。此外，由于开窗口的缘故，图像源不需要充满整个 FB ，因此每个象素可以同等对待。于是，对所有的 $j (1 \leq j \leq N)$ 来说，象素 $P(x, y)$ ，可以认为是方的，且大小相等，其中 j 代表第 j 的 FB ， N 是系统中帧缓冲器的数目。

总之，本发明论及并解决了在一个显示监视器中将多个图像源组合在一起的难题。对具有 N 个独立的图像源的系统提出了重叠和融合的技术。

本发明允许存在任何数目的独立图像源(N)，其中每一个图像源有一个相关的 FN 。该结构还能对每个 FB 的串行输出进行交错存取，以便与视频输出相适应。这样，如果交错存取需要 M 个矩阵，那么 $MIXER30$ 的总数是 $N \times M$ ，它能以一个象素的精度对任何数目的图像进行混合，其精度是图像的组合的透明系数的函数。

本发明还使用了视频查寻表，根据应用的特定窗口标识数，在一个任务环境中进行各种混合。

本发明还采用 WID 值，对每个窗口的任何数目的独立的图像

源，以逐个象素为基础，提供实时象素切换和控制。

虽然根据本发明的最佳实施例对本发明进行了展示和描述，但是本领域的技术人员应该懂得，在不偏离本发明的范围和精神的前提下，可以在形式上和细节上对其做各种改动。

说 明 出 图

图 1

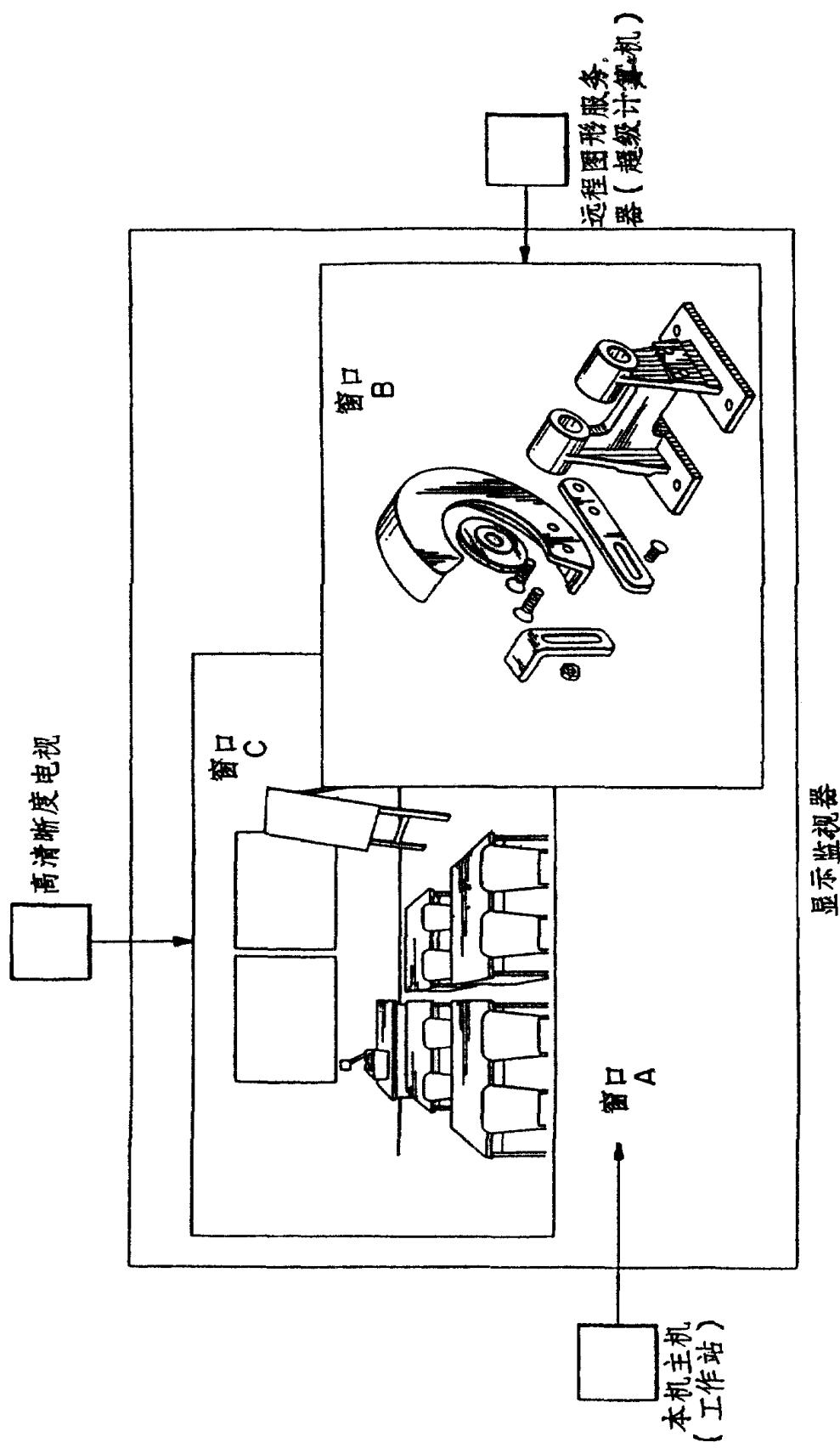


图. 2

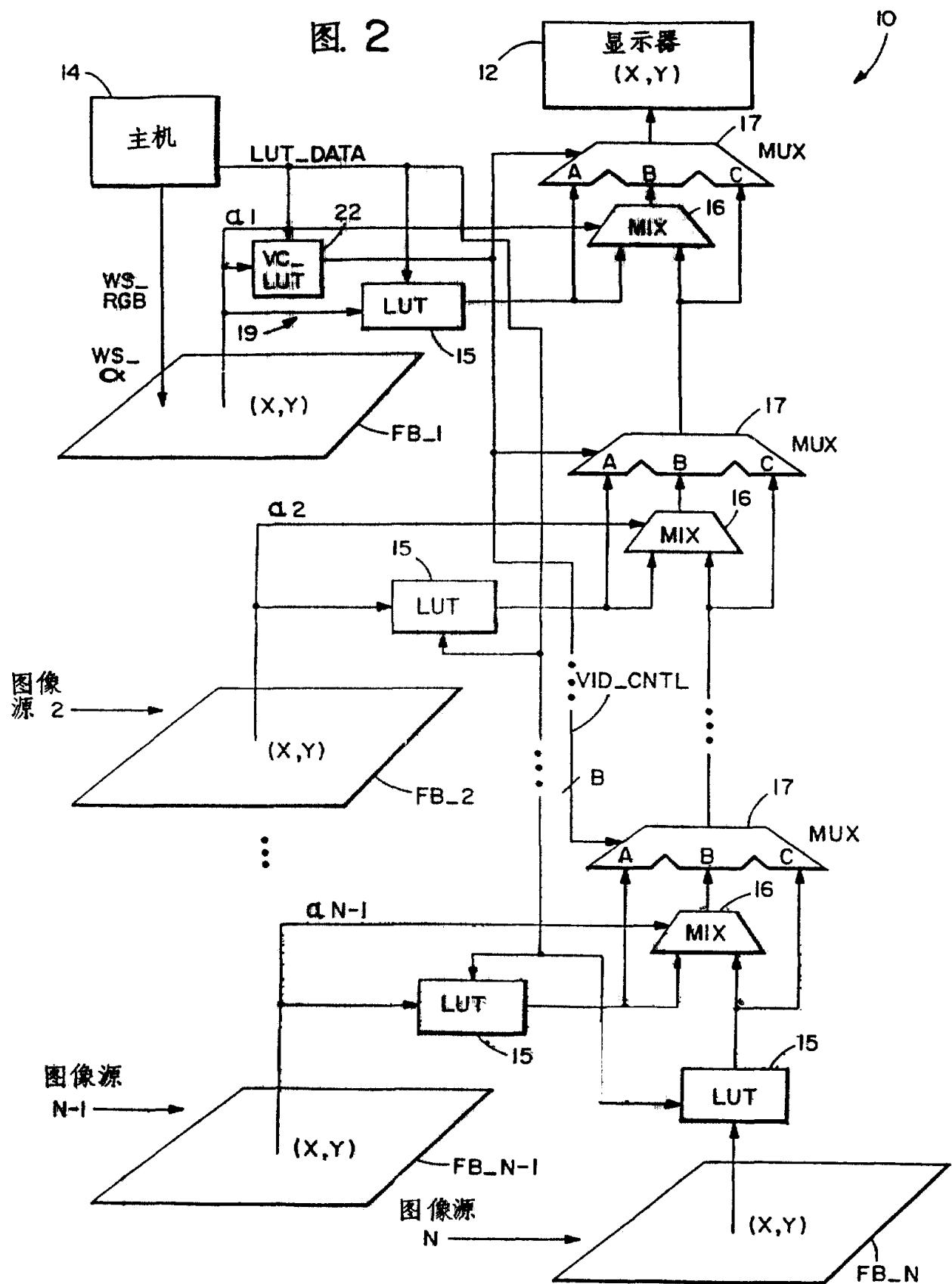


图.3
 $N \times M$ 结构

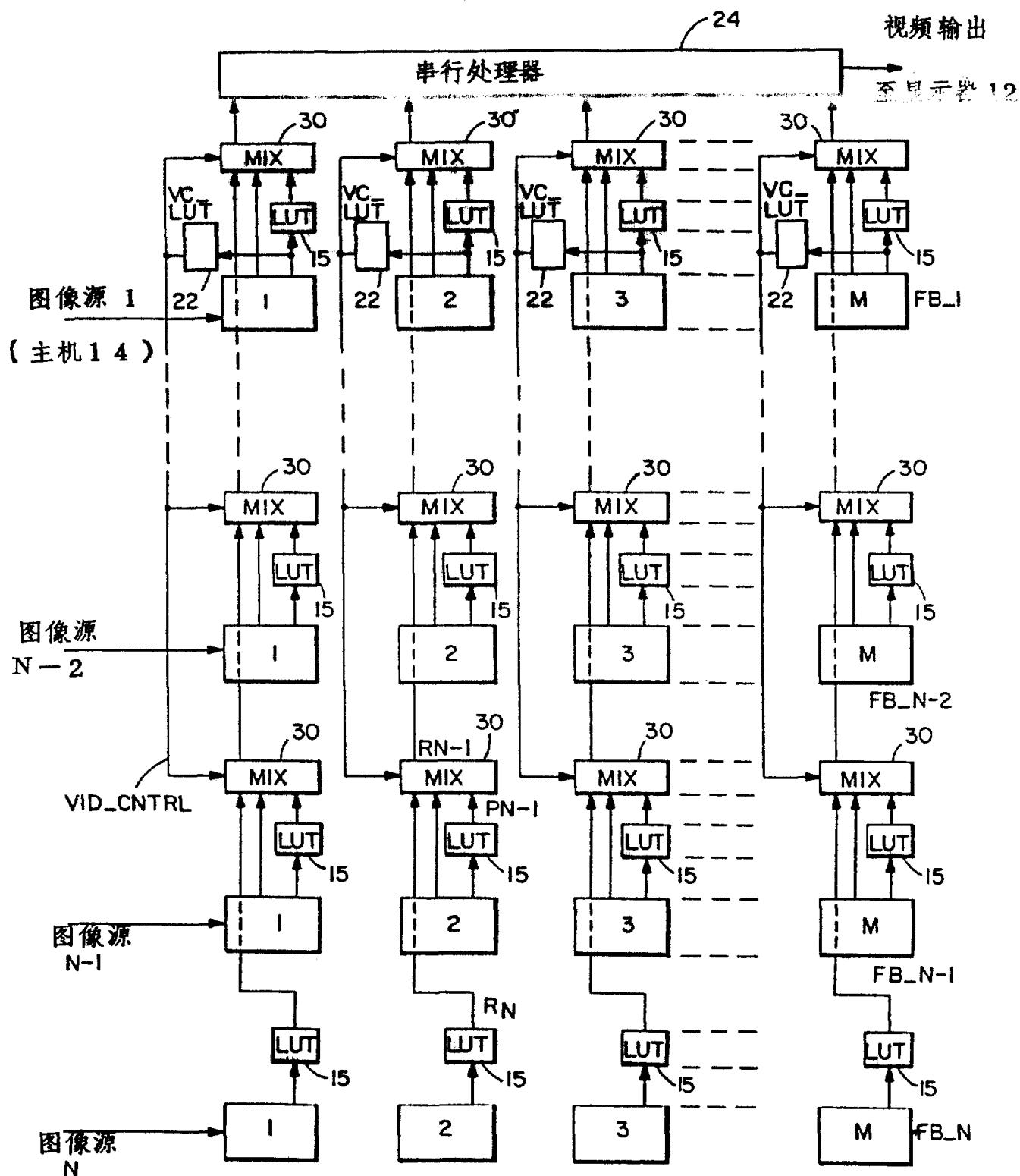


图. 4

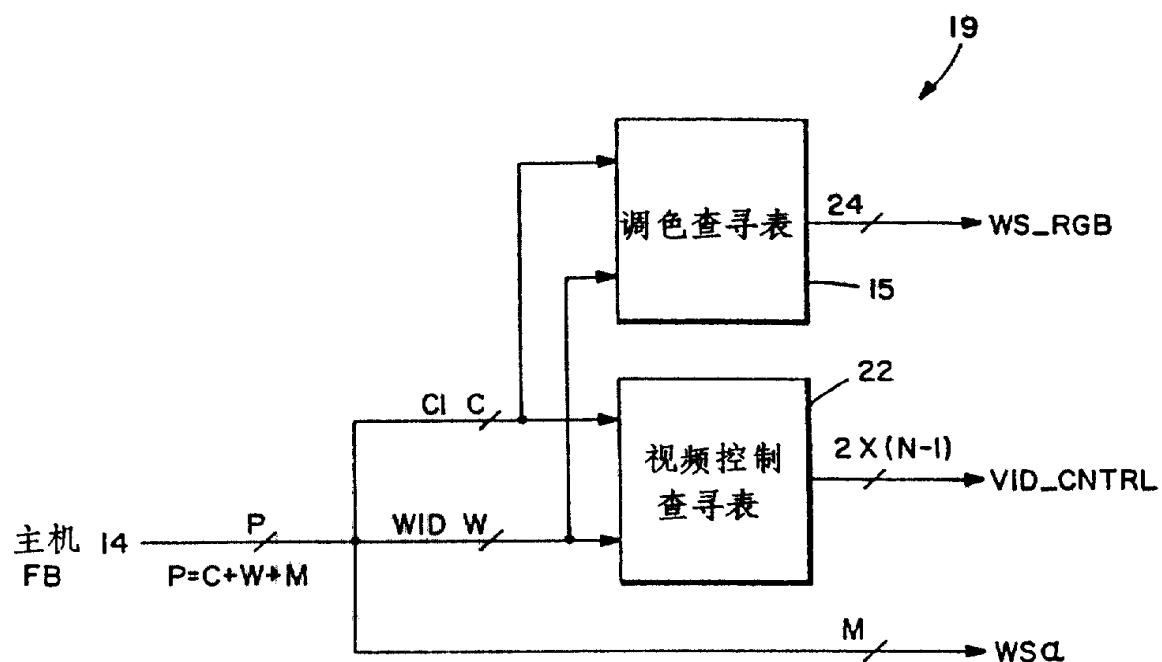


图. 6

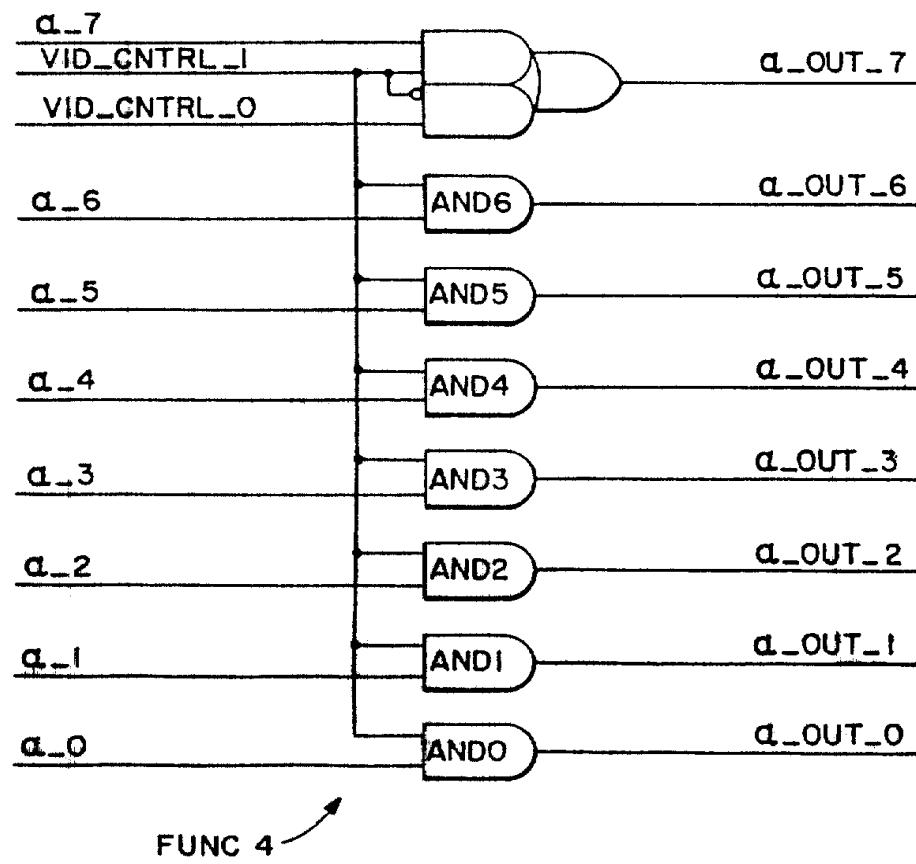


图. 5

