



[12] 发明专利申请公开说明书

[21] 申请号 96121310.8

[43]公开日 1997年9月17日

[11] 公开号 CN 1159619A

[22]申请日 96.11.12

[30]优先权

[32]95.11.13[33]US[31]555,456

[71]申请人 摩托罗拉公司

地址 美国伊利诺伊

[72]发明人 沃奥·博纳德·高勒 格日·林恩·米勒
大卫·里沃拉[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

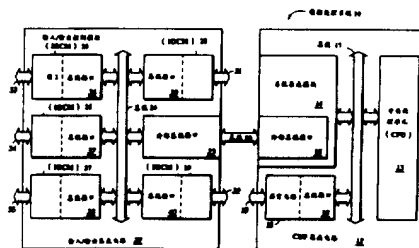
代理人 杨晓光

权利要求书 5 页 说明书 57 页 附图页数 26 页

[54]发明名称 改进计时器性能的集成电路输入/输出
处理器

[57]摘要

I/O 控制模块 (IOCM25—29) 具有通过计时器总线 (71, 72), 和管脚/状态总线 (75—77) 通信的通道。通道 (86, 87) 被每个计时器总线 (71, 72) 分成独立的通道块 (86, 87), 该通道块由它们各自的计时器总线 (71, 72) 提供对不同时基值的存取, 这样就不会损失分辨率, 因为在一个计时器块 (如 86) 中的每个通道, 都能从它对应的计时器总线 (71) 并行接收同样的时基值。管脚/状态总线 (75—77) 和计时器总线 (71, 72) 能被独立地划分。



权 利 要 求 书

1. 一种计时器处理器 (22), 其特征在于:

一个第一计时器总线 (71);

一个第二计时器总线 (72);

多个时基选择信号 (50);

一个第一时基通道 (80 中的一个), 用于产生一个第一时基值, 所述第一时基通道 (80 中的一个) 被耦合至所述多个时基选择信号 (50);

一个第二时基通道 (81 中的一个), 用于产生一个第二时基值, 所述第二时基通道 (81 中的一个) 被耦合至所述多个时基选择信号;

一个第一计时器总线控制通道 (61), 对应于所述第一时基通道 (80 中的一个), 用于选择所述第一和第二时基通道 (80, 81) 中的一个, 和如果所述第一时基通道 (80 中的一个) 被选择, 用于用第一时基值驱动所述第一计时器总线 (71), 所述第一计时器总线控制通道 (61) 被耦合至所述第一计时器总线 (71) 和所述多个时基选择信号 (50);

一个第二计时器总线控制通道 (62), 对应于所述第二时基通道 (81 中的一个), 如果所述第二时基通道 (81 中的一个) 被选择, 用于用第二时基值驱动所述第一计时器总线 (71), 所述第二计时器总线控制通道 (81 中的一个) 被耦合给所述第一计时器总线 (71) 和所述多个时基选择信号 (50);

一个管脚/状态总线, 用于提供管脚和状态信息;

一个管脚控制通道 (52), 用于控制在所述管脚/状态总线上, 管脚和状态信息的传输;

一个第一工作通道 (57), 被耦合至所述第一计时器总线 (71), 用于接收第一时基值, 和耦合至所述管脚/状态总线用于接收管脚和状态信息; 和

一个第二工作通道 (56), 被耦合至所述第二计时器总线 (72), 用于接收一个第三时基值, 和耦合至所述管脚/状态总线用于接收管脚和状态信息;

2. 一种集成电路计时器 (22), 其特征在于:

一个第一计时器总线 (71) ;

一个第一时基通道 (80 中的一个) , 用于产生一个第一时基值;

一个第二时基通道 (81 中的一个) , 用于产生一个第二时基值;

一个第一计时器总线控制通道 (61) , 对应于所述第一时基通道 (80 中的一个) , 用于选择所述第一和第二时基通道 (80 , 81) 中的一个, 和如果所述第一时基通道 (80 中的一个) 被选择, 用于用第一时基值驱动所述第一计时器总线 (71) , 和

一个第二计时器总线控制通道 (62) , 对应于所述第二时基通道 (81 中的一个) , 如果所述第二时基通道 (81 中的一个) 被选择, 用于用第二时基值驱动所述第一计时器总线 (71) 。

3. 一种集成电路计时器 (22) , 其特征在于:

多个数据导线 (435) ;

一个第一计时器通道 (400) , 其特征在于:

一个第一数据存储电路 (403) , 用于存储一个第一数据值;

一个第二计时器通道 (404) , 其特征在于:

一个第二数据存储电路 (404) ;

一个传输电路 (408) , 用于控制从所述第一数据存储电路, 到所述第二数据存储电路的, 第一数据值的一个传输, 第一数据值是通过所述多个导线传输的;

其中所述第一和第二计时器通道 (400 , 404) 中的第一个, 执行涉及第一数据值的一个计时器功能, 和其中所述第一和第二计时器通道 (400 , 404) 中的第二个, 存储第一数据值, 而不执行任何涉及第一数据值的计时器功能。

4. 一种计时器处理器 (22) , 其特征在于:

一个第一集成电路结合区 (110 中的一个) ;

一个第二集成电路结合区 (111 中的一个) ;

一个第一工作通道 (101) ;

一个第二工作通道 (102) ;

一个第三工作通道 (103) ;

一个第一管脚/状态总线，其特征在于：

一个第一管脚信息导线（112 中的一个）；和

一个第一管脚/状态信息导线（113 中的一个）；

一个第二管脚/状态总线，其特征在于：

一个第二管脚信息导线（114 中的一个）；和

一个第二管脚/状态信息导线（115 中的一个）；和

控制装置（105，106，107），用于通过所述第一管脚信息导线（112 中的一个），选择地在所述第一工作通道（101）和所述第一集成电路结合区（110 中的一个）之间传输的信息，用于通过所述第一管脚/状态信息导线（113 中的一个），选择地在所述第一工作通道（101）和所述第二集成电路结合区（111 中的一个）之间传输信息，用于通过所述第一管脚/状态信息导线（113 中的一个），选择地在所述第一工作通道（101），和所述第二工作通道（102）之间传输信息，和用于通过所述第一和第二管脚/状态信息导线（113，115），选择地在所述第一工作通道（101），和所述第二工作通道（103）之间传输信息，所述控制装置（105，106，107）被耦合到所述第一和第二集成电路结合区（110，111），所述第一和第二和第三工作通道（101，102，103），和所述第一和第二管脚/状态总线。

5. 一种集成电路计时器（22），其特征在于：

多个第一工作通道（206）；

一个第一局部总线（218），被耦合到每个所述第一多个工作通道（206）；

多个第二工作通道（205）；

一个第二局部总线（217），被耦合到每个所述第二多个工作通道（205）；和

一个全局通信总线（200），被选择地耦合到至少一个所述第一多个工作通道（206），和选择地耦合到至少一个所述第二多个工作通道（205），所述全局通信总线（200），从所述第一多个工作通道的至少一个（206）接收一个计时信号，和提供该计时信号给所述第二多个工作通道（205）的至少一个。

6. 一种集成电路计时器 (22), 其特征在于:

一个第一工作通道 (160);

一个第二工作通道 (161);

置位导线装置 (157), 用于从所述第一工作通道 (160) 接收一个第一置位值, 用于从所述第二工作通道 (161) 接收一个第二置位值, 和用于提供一个结果置位值, 所述置位导线装置 (157) 被耦合到所述第一和第二工作通道 (160, 161);

清零导线装置 (158), 用于从所述第一工作通道 (160) 接收一个第一清零值, 用于从所述第二工作通道 (161) 接收一个第二清零值, 和用于提供一个结果清零值, 所述清零导线装置 (158) 被耦合到所述第一和第二工作通道 (160, 161);

触发导线装置 (159), 用于从所述第一工作通道 (160) 接收一个第一触发值, 用于从所述第二工作通道 (161) 接收一个第二触发值, 和用于提供一个结果触发值, 所述触发导线装置 (159) 被耦合到所述第一和第二工作通道 (160, 161);

控制装置 (64), 用于接收结果置位值, 结果清零值, 和结果触发值, 所述控制装置 (64) 用于提供一个输出信号 (168), 这样输出信号的一个逻辑状态可被结果置位值, 结果清零值, 结果触发值所确定, 所述控制装置 (64) 被耦合到所述置位导线 (157), 所述清零导线 (158), 和所述触发导线 (159) 。

7. 一种集成电路计时器 (280), 其特征在于:

一个第一计时器模块 (281), 具有一个第一时基总线 (71, 或 271) 用于传输一个第一时基值;

一个第二计时器模块 (282), 具有一个第二时基总线, 用于传输一个第二时基值;

一个时钟导线 (328), 用于传输一个时钟信号, 该时钟信号由所述第一和第二计数器模块 (281, 282) 中的一个产生, 所述时钟模块 (328) 被耦合到所述第一和第二计数器模块 (281, 282); 和

一个同步导线 (329) 用于传输一个同步信号, 该同步信号由所述第一和第二计数器模块 (281 , 282) 中的一个确立, 所述同步导线被耦合到所述第一和第二计数器模块 (281 , 282);

其中所述第一计时器模块 (281) 特征在于:

一个第一时基计数器 (320), 用于产生第一时基值, 所述第一时基计数器 (320) 被时钟信号增加或减少, 当同步信号被确立时, 所述第一时基计数器 (320) 被装入一个预定值; 和

其中所述第二计时器模块 (282) 特征在于:

一个第二时基计数器 (320), 用于产生第二时基值, 所述第二时基计数器 (321) 被时钟信号增加或减少, 当同步信号被确立时, 所述第二时基计数器 (321) 被装入一个预定值。

8. 一种集成电路计时器 (22), 其特征在于:

一个第一计时器通道 (55);

一个第二计时器通道 (57);

计时器总线导线装置 (71), 用于在一个第一周期内传输一个第一时基值, 和用于在一个第二周期内传输一个第二时基值, 所述计时器总线导线装置 (71) 被耦合到所述第一和第二计时器通道 (55 , 57); 和

标志导线装置 (50), 用于传输对应于第一时基值的一个第一标志值, 和用于传输对应于第二时基值的一个第二标志值, 所述标志导线装置 (50) 被耦合到所述第一和第二计时器通道 (55 , 57) 。

说明书

改进计时器性能的集成电路输入/输出处理器

本发明一般涉及到输入/输出处理器，尤其涉及到具有改进计时器(timer)性能的输入/输出处理器。

复杂的实时控制系统，例如汽车和机器人控制系统，要求数据处理系统具有更好的控制，更快的系统响应和更多的输入/输出(I/O)能力。因此，对数据处理系统能进行更多，更快和改进输入/输出(I/O)处理的要求迅速增长。

人们期望微机能提供更高的数字和模拟的分解输出，以求更精确的控制。因为处理的额外开支，中央处理单元(CPU)可能不能以所需的速率响应甚高频控制功能。需要有一种解决方案，在不使CPU负担过重的前提下提供更多，更快和改进的输入/输出处理能力。

另外，人们期望微机可以非常灵活地适合各种不同应用。在以前的微机中，I/O处理的灵活性由中央处理单元(CPU)软件对端口或简单的计时器通道操作来提供。模拟和数字I/O也都通过CPU连接。然而，因为CPU涉及所有的I/O，I/O带宽主要浪费在系统的额外开支上了。这样，需要有一种解决方案同时提供灵活性和增进的I/O性能。

图1以框图的形式示出了根据本发明的一实施例的数据处理系统10；

图2以框图的形式示出了根据本发明的一实施例的图1中电路25的一部分；

图3以框图的形式示出了根据本发明的一实施例的图1中电路26的一部分；

图4以框图的形式示出了根据本发明的一实施例的图1中电路25的一部分；

图5以框图的形式示出了根据本发明的一实施例的图1中电路25的一部分；

图6以列表的形式示出了一张真值表，图5中管脚控制电路64使用该表判定结果导线168的逻辑电平；

图7以列表的形式示出了根据本发明的一实施例的图5中控制寄存器166的一部分和控制寄存器167的一部分；

图8以列表的形式示出了用户可以通过图7中管脚输出激励控制比特182或183编程实现的一组功能；

图9以列表的形式示出了在图5中管脚165通过匹配通道160和161状态的逻辑与和逻辑或所产生的结果示例；

图10以框图的形式示出了根据本发明的一实施例的图1中电路26和27的一部分；

图11以框图的形式示出了根据本发明的一实施例的图10中控制寄存器226的一部分；

图12以框图的形式示出了根据本发明的一实施例的图10中电路26的一部分；

图13以框图的形式示出了根据本发明的一实施例的图2中电路61，62，80，81和86中每个通道的一部分；

图14以框图的形式示出了根据本发明的一实施例的图2中电路25的一部分；

图15以列表的形式示出了用户如何在计时器总线的不同间隙中选择性地提供一个或多个时基值的8个例子；

图16以框图的形式示出了根据本发明的一实施例的一种计时器系统集成电路280；

图17以框图的形式示出了根据本发明的一实施例的一种数据处理系统315；

图18以框图的形式示出了根据本发明的一实施例的图16中主时基通道285的一部分和辅时基通道288的一部分；

图19以框图的形式示出了根据本发明的一实施例的图17中主时基通道304的一部分和辅时基通道310的一部分；

图20以框图的形式示出了根据本发明的一实施例的图1中电路25的一部分；

图21以框图的形式示出了根据本发明的一实施例的图20中捕获通道401的一部分；

图22以框图的形式示出了根据本发明的一实施例的图20中捕获通道401的一部分；

图23以列表的形式示出了置位和清零每一个通道的DVB比特（如图22中425）的行为，当该比特被配置到它的每一个操作模式时，可支持数据传送操作；

图24以列表的形式示出了进行数据传送操作的每一个通道的由数据传送控制比特（DTC）（例如图21和22中的423—424）控制的数据传送操作模式；

图25以分时图的形式示出了根据本发明的一实施例的执行周期累积测量的改进方法，其累积误差比现有技术的实现方法要少；

图26以分时图的形式示出了根据本发明的一实施例的执行周期累积测量的改进方法；

图27以框图的形式示出了根据本发明的一实施例的图2中计数通道58的一部分；

图28以框图的形式示出了根据本发明的一实施例的图2中寄存器67的第一部分；

图29以框图的形式示出了根据本发明的一实施例的图2中寄存器67的第二部分；

图30以框图的形式示出了根据本发明的一实施例的图2中寄存器67的第三部分；

图31以分时图的形式示出了根据本发明的一实施例的执行周期累积测量的改进方法；

图32以框图的形式示出了根据本发明的一实施例的图2中计数通道58的一部分；

图33以分时图的形式示出了根据本发明的一实施例的捕获时基值的改进方法；

图34以分时图的形式示出了根据本发明的一实施例的捕获时基值的改进方法；

图35以框图的形式示出了根据本发明的一实施例的图2中寄存器66的第一部分；

图36以框图的形式示出了根据本发明的一实施例的图2中寄存器66的第二部分；

图37以框图的形式示出了根据本发明的一实施例的图2中捕获通道55的一部分；

图38以框图的形式示出了根据本发明的一实施例的图2中寄存器65的第一部分；

图39以框图的形式示出了根据本发明的一实施例的图2中寄存器65的第二部分；

图40以框图的形式示出了根据本发明的一实施例的图2中匹配通道57的一部分；

术语“总线”是指传送一个或多个不同类型信息，如数据，地址，控制或状态信息的多个信号或导线。术语“取正”和“取负”各自指将信号，状态比特，集成电路管脚或类似设备再生成它的逻辑真或逻辑假状态。如果逻辑真状态是逻辑电平1，逻辑假状态则是逻辑电平0。如果逻辑真状态是逻辑电平0，逻辑假状态则是逻辑电平1。

术语“置位”是指将信号，状态比特，集成电路管脚或类似设备再生成它的逻辑电平1。术语“清零”是指将信号，状态比特，集成电路管脚或类似设备再生成它的逻辑电平0。术语“翻转”是指如果信号，状态比特，集成电路管脚或类似设备的目前状态是逻辑电平1，则再生成逻辑电平0。如果目前状态是逻辑电平0，则再生成逻辑电平1。

符号“%”前缀一个数字说明该数字以二进制形式表示，符号“\$”前缀一个数字说明该数字以十六进制形式表示。术语“集成电路管脚”和“管脚”可相互替代使用。另外，使用术语“集成电路管脚”和“管脚”处，也可以替换以结合区或任何在集成电路和外部设备间提供电气耦合的导线。

在以下描述中提出了许多具体的细节，如特定字和字节的长度等等以提供对本发明的彻底理解。然而，本领域的技术人员显然知道本发明可以在没有这些具体细节的情况下得到实施。在其他例子中，本发明采取电路框图的形式以避免不必要的细节。在绝大多数情况下，涉及定时考虑和类似情况的细节，因为不影响对本发明的完整理解，而且是相关领域中一般技术人员所熟悉的，因而被忽略。

尽管在本文件中通篇使用术语“集成电路管脚”和“管脚”（如图1中31—35和19），这些术语应该包括用于向某个集成电路传输电气信号，或接收电气信号的任何类型设备，例如，集成电路结合区，焊接块，导线等等。

现在参看图形，其中描述的元件不一定显示定标，并且相似元件在几张图中用同一个参数标出。

输入/输出控制模块（参看图1中IOCM 25—29）提供了数据处理系统的灵活性和增进的I/O性能。在本发明的一实施例中，一个或多个IOCM 25—29分布在与中央处理单元（CPU）13不同的集成电路中，但是一个或多个IOCM 25—29通过总线30与CPU 13通信。在本发明的可选实施例中，一个或多个IOCM可以与CPU处在同一个集成电路中。例如，参看图1，其它电路15可能作为输入/输出控制模块（IOCM）来实现，通过总线17与CPU 13通信。

IOCM 25—29的每一个设计成一个很模块化的系统结构。在最高层，IOCM 25—29的每一个是一个通过内部模块总线24与其它模块通信的模块。例如，在一实施例中，IOCM 25—29的每一个可以是一个用于MC68HC300族微机的模块，该类微机由Texas的Motorola Austin公司生产。

除了本身是一个模块，IOCM 25—29的每一个也由多个子模块或模块化I/O通道构成。通过任意组合“硅通道库”中的模块化通道，可以很容易地创建不同的IOCM版本。通道可以有不同的硬件，包括数据和模拟I/O电容，以完成特定操作。可以通过将新通道加入到库中的方式增加未预料的I/O功能。这样，合并不同的模块化通道布局就可以生成IOCM 25—29的不同版本。

IOCM 25—29的另一个重要性能是它们的结构允许用户在软件和硬件之间区分任务。通道可以编程在硬件中共同运行以完成简单的高频功能。功能运行的通道实际上充当高频I/O事件的预处理器。通过减轻CPU对所有I/O事件的软件服务，或仅要求CPU执行低频I/O事件的软件服务，实际上提供了更好的控制和更快的系统响应。

每一个IOCM 25—29的数字处理部分在所能执行的I/O功能的数量和性能上都是灵活的。因为在管脚，通道和总线的数量上是模块化的，每一

个IOCM 25—29都能实现这种灵活性。也就是说，不同IOCM版本可以从“硅通道库”中很容易地创建，集成电路管脚也可以增加，而与通道数量无关。当通道数量在不同IOCM版本中有所增加时，IOCM 25—29的这种结构也支持增加更多的数据和控制总线。可以设计新通道并加入到库中，以为将来的I/O功能提供一种增长方式。

IOCM 25—29系统结构的第二个重要方面是，通道可以编程，在硬件中共同运行，以实现简单的高频功能。在通过减少系统CPU中断数量而最大化系统输出时，允许用户在硬件和软件之间区分任务是重要的。

涉及同类型计时器功能的复杂实时控制系统需要大量的I/O处理。这样，相对小的通道库可以以不同的方式合并，组成多个IOCM 25—29以解决大多数用户的需求。

在本发明的一实施例中，通道库包括多个不同类别或类型的通道，即“工作通道”，“时基通道”和“其它通道”。这种类别命名将类似功能或结构的通道划分在一起。“工作通道”包括执行典型计时器功能的通道，如当时间事件发生时捕获一个输入值，当匹配事件发生时，提供一个信号和统计。工作通道也包括由基本计时器功能组成的更复杂的通道。例如，执行匹配和计数功能的通道可以统计匹配事件的数量，并仅在匹配事件数量达到某个预定值时提供一个信号。

在本发明的一实施例中，工作通道包括：（1）捕获通道（CC）；（2）匹配通道（MC）；（3）计数通道（CMTC）；（4）火花集成通道（SIC），执行多个计时器功能以提供发动机的火花定时；（5）燃料集成通道（FIC），执行多个计时器功能以提供发动机的燃料注入定时；（6）双向FIFO通道（DFC），提供一个存储数据值的2—深度FIFO。注意双向FIFO通道（DFC）不执行计时器功能，而代以提供数据存储功能。

在本发明的一实施例中，时基通道包括：（1）计时器总线控制通道（TBCC），提供计时器总线的主控制或辅控制；（2）度数时钟通道（DC），可提供多达四个时基；（3）计时器通道（TC），提供产生于内部或外部的时基；注意在本发明的一实施例中，计时器通道可以由一个或多个工作通道或一个或多个外部信号控制（如定时，加载某一模块化值，或由统计决定方向性）。

在本发明的一实施例中，其它通道包括：（1）执行同步序列转换的同步序列通道；（2）执行同步序列转换的异步序列通道；（3）管脚控制通道（PCC），负责工作通道与集成电路管脚的接口；（4）辅助外部总线接口通道（SEBI），负责内部模块间总线与该集成电路外部的总线之间的接口；（5）全局资源通道（GRC），包括不同版本间可能不同的客户化电路或指定电路（如特定计时器和一些系统保护特性）；（6）辅助总线接口通道（SBIU），用于与内部模块间总线接口；（7）主总线接口通道（MBIU），用于内部模块间总线（IMB）与度数计时器的接口（在CPU故障的情况下，所有的CPU写被限制只能写向一个预定的寄存器，在没有CPU的情况下，MBIU允许度数计时器接管IMB，并重新配置特定通道以运行系统，如汽车发动机）；（8）测试通道，包括用于测试目的电路。

注意在本发明的一实施例中，序列通道可以直接控制一个或多个相应集成电路管脚，也可以通过管脚控制通道使用相应集成电路管脚。

在本发明的一实施例中，可能存在其它不严格是通道库一部分的附加电路，但是这些电路以模块化的方式设计，可用以耦合一个或多个通道。例如，管脚控制共享逻辑可以耦合两个相邻管脚控制（PCC），允许两个相邻PCC共享总线信息。在本发明的可选实施例中，所有这种连接电路都可以作为通道库的一部分。

每一个工作通道提供许多可编程的属性，包括输入和输出事件。输入事件源可以是集成电路管脚或其它工作通道，而输出事件可以影响集成电路管脚和其它工作通道。通道操作，如输入捕获和相邻工作通道间的数据传输，可以由输入事件控制。相反地，工作通道的输出事件或集成电路管脚，可以影响工作通道的操作和工作通道间的数据传输。每一个工作通道的另一特性是可选的组态模式，该模式定义对输入事件或导致输出事件的操作所需执行的工作通道操作。这些特性和其它特性使工作通道可以编程共同运行，完成许多I/O功能。

通道库中的通道可以以不同方式添加，以创建IOCM 25—29的不同版本。在本发明的可选实施例中，可以有更多，更少或不同的通道库中的通道用于形成IOCM 25—29。通过使用模块化体系结构和灵活的模块化通道库，用户的许多I/O需求可以以一种迅速，有效的方式解决。

图1示出了数据处理系统10。数据处理系统10有一个CPU集成电路12和输入/输出(I/O)集成电路22。CPU集成电路12包括一个中央处理单元(CPU)13,系统集成模块14和其它电路15,都通过总线17双向耦合。系统集成模块14包括与总线30双向耦合的外部总线模块接口电路16。其它电路15包括总线接口电路18。其它电路15可以通过集成电路管脚19与CPU集成电路12外部耦合。

I/O集成电路22包括电路25—29和外部接口电路23,都通过总线24双向耦合。外部总线接口电路23双向耦合到总线30,向CPU集成电路12传送信息或从CPU集成电路12接收信息。I/O控制模块电路25—29通过集成电路管脚31—35与I/O集成电路22外部耦合。在本发明的可选实施例中,图1中的块31—35代表集成电路结合区而不是集成电路管脚。模块25—29的每一个包括一个或多个通道库中的通道(参看图2中的示例)。模块25—29的每一个也包括辅助总线接口通道(SBIU)36—40,用于实现一个或多个通道与内部模块间总线24的接口。

在本发明的可选实施例中,可能有一个或多个模块25—29。因为模块25—29内部总线负载的限制,将通道划分成一个或多个模块25—29,每一个模块拥有自己的总线接口36—40。每一个模块25—29中通道的数量由每一个模块25—29的内部总线的最终负载决定。体系结构的模块化允许每一个模块25—29可以由通道库中不同通道组成。这样,每一个模块25—29可以客户化以满足不同用户的需要。

模块化通道体系结构特性

除了使用模块化通道库之外,本发明也采用模块化方式实现通道总线,用以在不同的通道间传送信息。因为IOCM不同版本的通道布局不同,通道总线结构必须灵活。

参看图1和2,在本发明的一实施例中,IOCM(如IOCM 25)中的通道通过通道总线通信。下述通道总线可以在IOCM中实现通道的横向通信:

(1) 模块间总线(IMB)24的地址和数据导线;(2) 一个或多个计时器总线;(3) 一个或多个管脚/状态总线。通道也可以在两两之间建立互联通路以进行相邻通道的数据流动。模块间总线24中的地址和数据导线提供了

存取通道寄存器，控制寄存器和状态寄存器的方式。在本发明的一实施例中，总线接口电路36—40（参看图1）执行全局地址译码以判定特定IOCM 25—29是否被编址。然而，所有的本地地址译码都在每一个通道内完成。

计时器总线结构

参看图2，本发明使用一个或多个计时器总线71—72，可以简单地通过主计时器总线控制通道和辅计时器总线控制通道（TBCC）61—64将这些计时器总线划分成多个段，以增加可以向其它通道广播的时基的数量。

IOCM 25—29的体系结构和总线结构提供了沿计时器总线方向将计时器总线简单地划分成段的方法。其结果是，每一根计时器总线将通道划分成不同的通道块，通道块通过它们各自的计时器总线存取不同时基。因为计时器总线块内的每一个通道能够从计时器总线接收相同的时基值，所以—一个计时器总线块内的通道可以执行不同的功能而不损失分辨率。

作为例子，参看图2，注意计时器总线71对应于通道块中标以工作和其它通道86的通道块，计时器总线72对应于通道块中标以工作和其它通道87的通道块。计时器总线71用于向工作和其它通道86传送时基值，同样，计时器总线72用于向工作和其它通道87传送时基值。

注意每一个计时器总线71—72的一端由一个主计时器控制通道（TBCC）标出，计时器总线的另一端由一个辅计时器控制通道（TBCC）标出。例如，计时器总线71的一端由主计时器控制通道（TBCC）61标出，计时器总线71的另一端由辅计时器控制通道（TBCC）62标出。同样，计时器总线72的一端由主计时器控制通道（TBCC）63标出，计时器总线72的另一端由辅计时器控制通道（TBCC）64标出。主计时器控制通道 辅计时器控制通道用于标明计时器总线的每一部分或每一段。

参看图2，认识到计时器总线71和计时器总线72必须使用不同的导线是重要的，因为它们必须能同时传送不同的时基值。然而，计时器总线71和计时器总线72从概念上可以视为独立总线，或者也可以从概念上看成是同一个全局计时器总线的不同部分或段。在这个意义上，全局计时器总线是指向IC22的所有必须的工作和其它通道传送所有要求的时基值的总线。关键是对每一组必须同时传送的时基值，必须使用不同的计时器总线导线。

这些不同的导线可以被认为是独立总线，或者被认为是一根全局总线的不同部分或段。

在本发明的一些实施例中，主计时器控制通道 (TBCC) 可以单独控制计时器总线。辅计时器控制通道 (TBCC) 仅在时基通道块所提供的时基值需要在两根计时器总线上共享时才需要。另外，主TBCC也可以替代辅TBCC以允许两根计时器总线的时基值共享。例如，参看图2，如果计时器总线72仅需要从时基通道81来的时基值，可以不需要辅TBCC 64。辅TBCC 64用以从时基通道82向计时器总线72提供时基值。辅TBCC 64由主TBCC 63控制。

注意在本发明的可选实施例中，每一个通道块可以有更少，更多或不同的通道耦合到计时器总线。例如，工作和其它通道86和87可以包括更少，更多或不同的通道库中的工作通道或其它通道。

在包含模块化通道构造块的计时器系统中，分段的计时器总线结构提供了在沿计时器总线方向的任意处向计时器系统的不同部分方便地提供多个和不同时基的方式，并且不损失分解率。

例如，在一个典型的计时器系统中，工作和其它通道86可能需要以第一预定分解率存取特定时基，工作和其它通道87可能需要存取与通道86相关联的相同时基中的一些，再加上第二预定分解率的其它时基。本发明将全局计时器总线在需要向每一个IOCM 25—29输出的地方分段。其结果是，对应于每一个单个计时器总线的通道（如71和72）接收它们需要的时基以执行要求的功能，并且不损失分解率。

也是在需要可能非常长的计时器总线的计时器系统中，因为需要支持大量的功能，本发明提供了将计时器总线分段的方式，以保证计时器总线的每一部分具有足够的驱动能力。

在本发明中，计时器总线从时基通道中（如图2中时基通道80中的一个）接收一个时基值，并向一个或多个工作和其它通道（如图2中工作和其它通道86）提供该时基值。通过同时向多个工作和其它通道提供时基值，在工作和其它通道之间保持一致。

在本发明的一实施例中，计时器总线是一个时分多路复用总线，允许多达8个时基通道提供多达8个不同时基值，在同一个计时器总线中时分多

路复用。在本发明的可选实施例中，计时器总线可能不是时分多路复用的，或者时分多路复用成不同数量的时间段。

参看图2，在本发明的一实施例中，主计时器总线控制通道（TBCC）61选择时基通道80和81中哪一个或哪几个可以提供时基值以驱动计时器总线71。如果主TBCC 61选择时基通道80中的一个，例如计时器通道91，则主TBCC 61以计时器通道91所提供的时基值驱动计时器总线71。但如果主TBCC 61选择时基通道81中的一个，例如计时器通道92，则辅TBCC 62以计时器通道92所提供的时基值驱动计时器总线71。这样，只有主TBCC选择哪个计时器通道提供时基值，但是主TBCC和辅TBCC都可以从时基通道接收时基值，并以该时基值驱动计时器总线。

在本发明的一实施例中，每一个主TBCC提供对相应辅TBCC的所有控制。在每一个主TBCC中的控制寄存器存储比特（如TBCC 61中的控制寄存器存储比特 68和TBCC 63中的控制寄存器存储比特 69）用于在计时器总线的8个时分多路复用窗口的每一个活跃期间，控制激活某个时基选择信号。从主TBCC来的时基选择信号用于选择激活的时基，通过主TBCC（从时基通道80）或辅TBCC（从时基通道81）选通到相应的时分多路复用计时器总线71。

参看图2，时基选择信号50必须从主TBCC 61延伸到与主TBCC 61相关的时基通道80，并从主TBCC 61跨越工作和其它通道86到辅TBCC 62和它的相关时基通道81。除此之外，时基选择信号50还被工作和其它通道86用于判断哪个时基在计时器总线上是物理活跃的，以确定何时执行诸如匹配和捕获的特定操作。这样，时基选择信号50有效地从主TBCC 61向工作和其它通道86和时基通道80，时基通道81传送了一个标识值。该标识值由时基选择信号50传送，以指示在计时器总线71上哪个时基目前可用。

注意，在本发明的一实施例中，图2中示出的每一个信道有一个或多个用户可编程通道控制寄存器260（参看图13）。不同类型的通道可以有一些相同的寄存器。参看图2和图13，在本发明的一实施例中，每一个80，81和86中的通道以及每一个通道61和62都有一个或多个如图13所示的耦合的通道寄存器260。

参看图13，通道控制寄存器260的一部分264存储用户编程标识值，该值可选择进行编码或不编码。如果标识值进行编码，该编码标识值提供给

译码电路261，该电路在输出上提供一个译码标识值。该译码标识值随后与时基选择信号50提供的标识值比较。如果该标识值匹配时基选择信号50上驱动的标识值，则匹配信号263被取正以向该通道指示该通道或者向计时器总线提供一个时基值（对时基通道而言），或者从计时器总线接收时基值（对工作和其它通道而言）。

在本发明的可选实施例中，该标识值存储在通道控制寄存器260的264部分，未进行编码。其结果是不再需要译码器261。取代的是，264部分直接作为多路复用器（MUX）电路262的输入。264部分控制MUX电路的输入，选择时基选择信号50中的一个，在输出时提供如匹配信号263。

本发明的计时器总线结构支持下述功能。首先，计时器总线结构允许沿其长度方向的任意处分段。其次，允许一对计时器总线控制通道（TBCC），即主TBCC和辅TBCC，控制计时器总线的分段。第三，允许相同时基通道（例如图2中的时基通道81）同时向两个不同计时器总线段（如计时器总线71和72）提供相同时基通道。

第四，本发明的计时器总线结构允许每一个计时器总线划分成N个时分多路分时间窗口。第五，计时器总线结构支持用户编程选择M个时基值中的任意一个，在每一个计时器总线段中的N个分时间窗口中的每一个中驱动。例如，图2中时基通道81可以包括不同数量的时基通道（例如计时器通道92和计时器通道95）。每一个时基通道可以向计时器总线71，计时器总线72或同时向计时器总线71和72提供一个或多个时基值。数字M和N是正整数。

作为例子，参看图2，在本发明的一实施例中，时基通道81可以提供5个时基值（也就是M=5），其中计时器通道92提供2个时基值“A”和“B”，计时器通道95提供3个时基值“C”，“D”和“E”；计时器总线71和计时器总线72可以每一个都时分复用成8个分时间窗口（也就是N=8）。计时器总线71的8个分时间窗口可以驱动下列时基值：A，D，A，B，A，D，A，C，当然在计时器总线71的下一组8个分时间窗口中重复。在另一方面，计时器总线72的8个分时间窗口可以驱动下列时基值：B，D，B，D，B，D，B，D，当然在计时器总线72的下一组8个分时间窗口中重复。在第二和第六个分时间窗口中，计时器总线71和计时器总线72驱动同一个由计时器通道95提供的时基值“D”。

图15示出了8个关于用户怎样在计时器总线的不同时段间选择性提供一个或多个时基值（即TB1, TB2, TB3, TB4, TB5, TB6, TB7, TB8）的例子。每一个时基通道80提供时基值TB1—TB8中的一个。每一个时基通道80由用户编程寄存器存储电路264（参看图13）存储标识值。计时器选择信号50在每一个时段中驱动一个标识值，该标识值匹配时基通道中的编程标识值，时基通道驱动时基源总线271（参看图14），从而以该时基值（即时基值TB1—TB8中的一个）驱动了计时器总线71。

注意在本发明的可选实施例中，一个时基通道针对每一个时基值（TB1—TB8）有一组可编程寄存器存储比特，该通道可以向计时器总线71提供这些比特。每组可编程寄存器存储比特264可以编程对存储比特264执行写操作（如从图1中的CPU 13）。在本发明的可选实施例中，标识存储比特264可以屏蔽可编程性，这样它们可以在I/O集成电路22生产期间一次编程成某个固定值。

注意除了时基通道81，时基通道80也可以由用户编程，为计时器总线71提供一个或多个时基值。同样，除了时基通道81，时基通道82也可以由用户编程，为计时器总线72提供一个或多个时基值。这样，可供选择并在计时器总线71上驱动的时基值的总数量是由时基通道80和时基通道81提供的时基值的总和。同样，可供选择并在计时器总线72上驱动的时基值的总数量是由时基通道81和时基通道82提供的时基值的总和。

本发明的计时器总线结构还有更多的性能。第六，本发明的计时器总线结构通过主计时器总线控制通道和辅计时器总线控制通道（如图2中时基通道61和时基通道62）控制多个时基值选通到通常的计时器总线段（如计时器总线71）。时基选择信号50包括在时基通道80和时基通道81中的每一个时基通道上的一个选择信号。通过在适当的分时窗口期间取正时基选择信号中的某一个，主时基通道61选择一个时基通道选通到计时器总线71。

第七，本发明的计时器总线结构允许工作和其它通道86监控时基选择信号50，以判定时基通道中的哪一个目前正向计时器总线提供时基值。其结果是，工作和其它通道86的每一个可以决定何时执行它们各自的操作，如匹配或捕获计时器总线71上的被选时基值。工作和其它通道86在由时基选择信号50决定的适当时候读计时器总线71，并使用从计时器总线71读来的时基值执行操作。

时基通道和时基同步

参看图2，本发明使用一个或多个时基通道（如81）以产生时基值，通过计时器总线（如71，72）向工作和其它通道86和87提供。IOCM 25—29的体系结构和总线结构（参看图1）将每一个IOCM划分成不同的工作和其它通道块（如85—87），这些通道块可以存取不同时基通道（如80—81）。因为每一块中的每一个工作和其它通道可以从计时器总线中接收相同时基值，所以在通道块中的工作和其它通道可以用于执行不同功能而并不损失分解率。

在本发明的一实施例中，时基通道（如80，81）可以包括下列的一个或多个：（1）计时器总线控制通道（TBC），提供计时器总线的主控制或辅控制；（2）度数计时器通道（DC），可提供多达四个时基；和（3）计时器通道（TC），提供产生于内部或外部的时基。

参看图2和图14，在本发明的一实施例中，时基通道80的每一个与时基源总线271耦合。时基源总线271以与计时器总线71相同的方式进行时分复用。时基通道80的每一个比较时基选择信号和存储在寄存器部分264的用户编程标识值（参看图13）。如果匹配，该匹配通道以通道时基驱动时基源271。不匹配的通道在这个特定时间间隔内不驱动时基源总线271。耦合逻辑270（参看图14）用于以合适的发送次数从时基源总线271向计时器总线71提供下一个时基值。

出于计时器总线的驱动能力，在计时器总线上的数据传送频率，不同计时器系统物理位置或路由方面的考虑，一个复杂的数据处理系统（如图17中的数据处理系统315）如果太大，就可能无法在单个时基内完成其功能。在一些数据处理系统中，在数据模块之间划分计时器系统是必要的（参看图16）。因为整个系统的规模，或者因为不同技术的要求，如功率输出驱动，模拟输入条件和复杂的数字电路，在多个集成电路之间进行划分是必要的。参看图17，在接口集成电路和功率集成电路现有的打包技术下，向接口集成电路301或向功率集成电路302分布一个16比特或32比特时基值是不可行的。

参看图16—19, 本发明仅通过两个信号, 一个计时器信号328和一个同步信号329, 使两个或多个时基通道 (如图16和图18中的主时基通道285和辅时基通道288) 同步或保持同步。计时器信号328和同步信号329可以在不同集成电路的计时器系统间 (参看图17), 或在一个集成电路的不同模块间 (参看图16) 进行广播。主时基通道产生或接收主计时器信号, 连接一个或多个辅计时器信道以保证主时基通道和辅计时器信道同时以相同的速率增长或递减。

例如, 在图18中, 主时基通道 285通过计数系统计时器信号327产生一个主计时器信号328。主时基通道 285随后将主计时器信号328提供给一个或多个辅时基通道 (如288) 以保证主时基通道和辅计时器信道同时以相同的速率增长或递减。在图19示出的本发明的一个可选实施例中, 主时基通道304和辅时基通道310都接收相同的主计时器信号348, 主计时器信号在本发明的一些实施例中, 是与系统计时器327相同的信号 (参看图18)。然而, 主时基通道304和辅时基通道310都以相同数量计数主计时器信号348。这样, 时基递减计数器340和时基递减计数器341同时以相同的速率递减。

如果数据处理系统 (如图17中的315) 使用一个固定频率的系统计时器, 并且该系统计时器可以作为主时基通道的输入和一个或多个辅时基通道的输入, 那么主要在主时基通道和辅时基通道的预定标电路326, 346和350 (参看图18和图19) 中选择相同的预定标分值, 该系统计时器即可用作主计时器信号。这实际上节省了必须增加的一根新的主计时器信号互联导线。在使用非时域或不规则频率的系统计时器信号的数据处理系统中, 主时基通道必须产生主计时器信号。关键在于所有的同步时基通道 (如图18中的285和288, 图19中的304和310) 必须以相同频率和相位的主计时器信号 (如图18中的328和图19中的348) 定时。

然后需要同步信号 (如图18中的327和图19中的349) 将所有时基通道 (如图18中的285和288, 图19中的304和310) 设置成相同的初始值, 在此基础上由上述主计时器信号进行增量/减量。对使用自激计时器的时基通道, 最方便的同步点是到\$0000或\$FFFF的翻转点, 取决于它是一个递增计数器还是一个递减计数器 (参看图18)。对模数时基而言, 模数值被装入计数器的点是最方便的同步点 (参看图19)。

选择这些同步点是因为主时基通道（如图18中的285，图19中的304）已经有了所需的用于其它用途的检测电路（如图18中的330，图19中的351）。另外，在这些被选同步点中的时基是所有辅时基通道中已有的或简单产生的（如递减计数器的\$0000，递增计数器的\$FFFF和模数计数器中的模数值）。主时基通道用于检测同步点的出现以产生同步信号。同步信号随后提供给一个或多个辅时基通道，其中同步信号用以同时将所有时基值重置成相同值。

一旦在每一个时基通道中的计数器（图18中的320和321，图19中的340和341）已被设成相同值，并以相同频率和相位主计时器信号定时，计数器应该保持所有计数值的完全同步。如果一个时基值因为噪音或其它影响而失去同步，在下一个同步点由同步信号进行重新同步。

这样，本发明允许在不同I/O控制模块（IOCM）281—284（参看图16）或不同集成电路300—302（参看图17）上的时基通道产生和使用一个同步化和一致的时基值。因为电路负载的限制，可以与同一个计时器总线71耦合以从时基通道80和81接收时基值的工作和其它通道数量的存在一个上限值。这样，本发明允许不同IOCM和不同集成电路上的时基通道向与不同计时器总线耦合的不同的工作和其它通道提供同步化和一致的时基值。在使用最小数量的集成电路管脚时，向外设集成电路（如图17中的301和302）增加计时器通道的性能是重要的。

参看图16，注意在本发明的一实施例中，在计时器系统集成电路280中的IOCM 281包括图2示出的所有电路。同样，在计时器系统集成电路280中的IOCM 284包括图2示出的所有电路的副本。不同于模块间总线24（参看图1），全局通道通信总线200（参看图10），主计时器信号328和同步信号329，IOCM 281和IOCM 286不需要任何其它导线在其间耦合以传送信息。

注意在同步时基通道的上下文中，术语“主”和“辅”用于指示哪个时基通道提供同步信号（主）和哪个时基通道接收同步信号（辅）。术语“主”和“辅”与主计时器总线控制通道和辅计时器总线控制通道的意义不同，与主管脚控制通道和辅管脚控制通道的意义也不同。然而，一般说来，术语“主”指出提供较多控制的电路，术语“辅”指出提供较少控制或从主电路那儿接收较多控制信号的电路。

管脚/状态总线结构

参看图1和图2, 管脚/状态总线75—77, 与计时器总线71和72类似, 是模块化化的, 可以扩展或分段以创建运载不同信号的独立总线。本发明使用一个或多个管脚/状态总线 75—77, 这些总线可以通过管脚控制 (PCC) 简单地划分成不同的总线和段。

管脚/状态总线75—77用于在IOCM 25内的通道间传送信息。在本发明的一些实施例中, 一个或多个管脚/状态总线 (如77) 用以从一个IOCM (如25) 向另一个不同的IOCM (如26) 传送信息。另外, 管脚/状态总线75—77用于在集成电路管脚33和IOCM 25的一个或多个通道间传送信息。集成电路管脚31—35用于接收和提供I/O集成电路22的外部信息。每一个IOCM 25—29都有一个或多个管脚/状态总线以在多个IOCM通道间或IOCM通道与一个或多个集成电路管脚31—35间传送信息。

在本发明的一些实施例中, 管脚/状态总线75—77充当通道影响和控制耦合到相同管脚/状态总线上的其它通道的方式。管脚/状态总线75—77也充当通道影响和控制输出集成电路管脚逻辑电平的方式, 以及通道被输入集成电路管脚的逻辑电平影响和控制的方式。

在本发明的一实施例中, 管脚/状态总线的每一个 (如图2中75—77) 可以用作四个用途: (1) 指示编程输入的一个或多个集成电路管脚的逻辑电平; (2) 决定编程输出的一个或多个集成电路管脚的逻辑电平; (3) 充当一个或多个通道的输入事件源; (4) 充当一个或多个通道的输出事件目的地。

由CPU 13 (参看图1) 协调向通道的数据写入是必须的。在使用管脚/状态总线的情况下, 一致性必须得到保证。这些总线提供了通道间的同步控制, 所以通道操作可以相互同步。因为CPU 13也通过管脚控制中的控制寄存器影响这些总线, 所以通道操作可以与CPU 13操作同步。其结果是, 通道数据的一致存取可以由CPU 13对管脚/状态总线的控制得到保证。

IOCM 25—29的体系结构和总线结构提供了沿长度方向的任意处将管脚/状态总线简单地划分成段的方法。其结果是, 通道被划分成可以存取不同集成电路管脚的不同通道块。在一个通道块中的每一个通道可以向多

个，一个，或没有的集成电路管脚提供信息，或从多个，一个，或没有的集成电路管脚接收信息。同样，在一个通道块中的多于一个通道可以向同一个集成电路管脚提供信息，或从同一个集成电路管脚接收信息。参看图2，注意将通道划分成能够存取不同管脚/状态总线75—77的不同通道块与将通道划分成能够存取不同计时器总线的不同通道块无关。

作为例子，参看图2，注意管脚/状态总线75对应于标以57的通道块，管脚/状态总线76对应于标以58的通道块。管脚/状态总线75用于在通道57间传送管脚和/或状态信息，同样，管脚/状态总线76用于在通道58间传送管脚和/或状态信息。

与计时器总线控制通道（如图2中61—64）不同，管脚控制不区分“主”和“辅”版本。取代的是，每一个管脚控制负责控制它相应的管脚/状态总线。例如，在图2中，管脚控制51控制管脚/状态总线75；管脚控制52控制管脚/状态总线76；管脚控制53控制管脚/状态总线77。

然而，如图3中所示，管脚控制共享逻辑106可用以选择性地将管脚/状态总线118的一个或多个导线耦合到管脚/状态总线119的一个或多个导线。这样，实际上组成了一个扩展的管脚/状态总线（118，119）。例如，管脚控制共享逻辑106可以联结管脚/状态总线118和管脚/状态总线119，这样，总线118和119的一个或多个导线可以传送相同的数值或信号。其结果是，管脚控制共享逻辑106允许管脚/状态总线118和119的一部分或全部进行有效连接或扩展成一根总线，该总线在实际上两倍的通道和两倍的集成电路之间运载相同的信号，传送相同的信息。

参看图3，图3中示出的IOCM 26部分包括4个管脚/状态总线116—119。管脚/状态总线116由管脚控制109控制，不向任何其它管脚/状态总线传送任何信息。管脚/状态总线117由管脚控制108控制，不向任何其它管脚/状态总线传送任何信息。管脚/状态总线118由管脚控制105控制，并通过管脚控制共享逻辑106向和/或从管脚/状态总线119传送信息。管脚/状态总线119由管脚控制107控制，并通过管脚控制共享逻辑106向和/或从管脚/状态总线118传送信息。

在本发明的一实施例中，每一个管脚/状态总线（如118）有8根导线用以传送管脚信息（如112），8根导线可以单独编程以传送8个管脚信息或状态信息（如113）。在管脚控制105中的控制寄存器存储比特120用于选择性

地决定，对于每一个管脚/状态总线113，该导线是否在通道123和一个或多个集成电路管脚110之间传送管脚信息，或是否该导线仅在通道123间传送状态信息。同样，在管脚控制107中的控制寄存器存储比特122用于选择性地决定，对于每一个管脚/状态总线115，该导线是否在通道125和一个或多个集成电路管脚111之间传送管脚信息，或是否该导线仅在通道125间传送状态信息。

在管脚控制共享逻辑106中控制寄存器存储比特的一部分用于控制管脚/状态总线118的导线与管脚/状态总线119的导线的选择性耦合。在本发明的可选实施例中，管脚控制共享逻辑106中控制寄存器存储比特的一部分可以复制，作为管脚控制（PCC）105和107物理电路的一部分。可以改变屏蔽选项以激活两个管脚控制105和107中的一个的控制寄存器存储比特的这一部分。

例如，改变屏蔽选项可以激活管脚控制 105中的控制寄存器存储比特121的一部分，而管脚控制 107中控制寄存器存储比特121的复制部分保持未激活状态。管脚控制 105中激活的控制寄存器存储比特121部分与管脚控制共享逻辑106耦合，并用于控制管脚控制共享逻辑106，而管脚控制107中未激活的控制寄存器存储比特121的复制部分不与管脚控制共享逻辑106耦合，对管脚控制共享逻辑106无影响。在本发明的一实施例中，具有部分激活的控制寄存器存储比特121的管脚控制105被认为是“主”管脚控制，而具有部分未激活的控制寄存器存储比特121的管脚控制107被认为是“辅”管脚控制。

参看图2，认识到管脚/状态总线75—77必须使用不同的导线是重要的，因为它们必须能同时传送不同组的管脚/状态值。然而，管脚/状态总线75—77从概念上可以视为独立总线，或者也可以从概念上看成是同一个全局管脚/状态总线的不同部分或段。在这个意义上，全局管脚/状态总线是指向集成电路22的所有必须的通道和所有必需的管脚传送所有要求的管脚/状态值的总线。关键点是对每一个必须同时传送的管脚/状态值组，必须使用不同组的管脚/状态总线导线。这些不同的导线可以被认为是独立总线，或者被认为是一根全局总线的不同部分或段。

注意在本发明的可选实施例中，每一个通道块可以有更少，更多或不同的通道耦合到管脚/状态总线。例如，参看图2，通道57和58可以包括更少，更多或不同的通道库中的可用通道。

在包含模块化通道构造块的计时器系统中，分段的管脚/状态总线结构提供了在沿管脚/状态总线方向的任意处向计时器系统的不同部分方便地提供多个和不同管脚/状态信息的方式，并且不损失分解率。

例如，在一个典型的计时器系统（参看图2）中，通道58之间可能需要交换信息，并在管脚控制52控制下存取集成电路管脚（没有示出），而通道57之间可能需要交换信息，并在管脚控制51控制下存取集成电路管脚（没有示出）。参看图3，如果一个或多个通道123需要提供信息，或从一个或多个通道125接收信息，管脚控制共享逻辑106必须插入到管脚控制105和管脚控制107之间以允许管脚/状态总线118和管脚/状态C119共享信息，并同时传送相同的管脚/状态值。本发明将全局管脚/状态总线在需要向每一个IOCM 25—29输出的地方分段。其结果是，对应于每一个单个管脚/状态总线的通道（如75，76和77），接收和提供执行要求的功能所需要的管脚/状态信息。

也是在需要可能非常长的管脚/状态总线的计时器系统中，因为需要支持大量的功能，本发明提供了将管脚/状态总线分段的方式，以保证管脚/状态总线的每一部分具有足够的驱动能力。

灵活的管脚与和或输出

图3中示出的管脚/状态总线的特定实施例仅仅是一种可能的实施例，还有许多其它的实施例。图4中示出的管脚/状态总线的另一个可选实施例，允许用户逻辑合并多通道输出以判断输出管脚的状态。

在图4中示出的实施例中，图1中管脚/状态总线75—77的每一个各自由输入状态总线143—145和输出事件总线131—133实现。每一根输入状态总线143—145各自包括8根管脚导线146，148和150；以及8根管脚/状态导线147，149和158。每一个输出事件总线131—133各自包括8根置位导线134，137和140；8根清零导线135，138和141，以及8根翻转导线136，139和142。

输出事件总线132用于判断管脚195的输出状态，对应于和被控于管脚控制通道52。通道58中的不同通道通过输出事件总线影响管脚195的逻辑状态。管脚195的每一个耦合到置位导线137中的某一个，清零导线138中的某一个，和翻转导线139中的某一个。这样，特定管脚195的逻辑状态由耦合到该特定管脚的这三根导线的逻辑状态决定，即置位，清零和翻转。然而，注意置位，清零和翻转导线不影响由管脚控制通道52配置为输入管脚的管脚195。与管脚相关联的置位，清零和翻转导线仅在该管脚被配置为输出管脚时才起作用。

仍参看图4，在本发明的一实施例中，24根导线137—139的每一个都是一个能够与其它导线进行线接异或的线接异或导线。在本发明的一实施例中，一个工作通道块（如160，161）中的所有置位导线137是共同线接异或的。同样，一个工作通道块（如160，161）中的所有清零导线138是共同线接异或的。类似的，一个工作通道块（如160，161）中的所有翻转导线139是共同线接异或的。

作为示例，以下描述管脚/状态总线76（参看图4）的操作。图5示出了管脚/状态总线76的一部分，在管脚165被配置为输出管脚时，用于控制集成电路管脚165的输出状态。置位导线157是图4示出的8根置位导线137中的一个；清零导线158是图4示出的8根清零导线138中的一个；翻转导线159是图4示出的8根翻转导线139中的一个。每一根导线157—159是一根预充电至逻辑电平1的线接异或导线。这样，如果匹配通道160的管脚控制电路162在导线171上驱动逻辑电平1，而匹配通道161的管脚控制电路163在导线168上驱动逻辑电平0，置位导线157的逻辑电平将是逻辑电平0。因此，对线接异或导线而言，逻辑电平0是主要的。可选实施例也可以使用线接或导线。

管脚控制电路164接收置位导线157，清零导线158，翻转导线159的线接异或结果。管脚控制电路164也通过导线174接收管脚165的当前逻辑电平。管脚控制电路164使用导线157—159和174的逻辑电平以判定导线168的逻辑电平结果，进而判定集成电路管脚165的下一个输出逻辑电平。

图6示出了由管脚控制电路164的一个实施例所用的真值表，以判定导线168的逻辑电平结果，进而判定集成电路管脚165的下一个输出逻辑电平。结果导线168用于驱动管脚165的输出逻辑电平。图6中示出的真值表定义了一个协议，用于决定管脚165对置位，清零和翻转导线157—159的所有

可能情况所采取的行动。对图6的表而言，置位，清零和翻转导线157—159的正状态是逻辑电平“0”，置位，清零和翻转导线157—159的负状态是逻辑电平“1”。该协议可以陈述如下：（1）如果翻转导线159是正状态，翻转总成功，管脚165被翻转；（2）如果所有导线157—159是负状态，管脚165保持原电平；（3）如果置位导线157和清零导线158都是正状态，翻转导线159是负状态，管脚165保持原电平。可选实施例可以使用不同协议和不同真值表。

通过带有独立置位（137）、清零（138）和翻转（139）导线的输出事件总线132，本发明能够在多个通道的输出上执行逻辑操作，而不需要服务处理器介入。这是非常强大和灵活的性能。

为初始化计时器输出管脚逻辑电平的改变以作为多个计时器通道的结果状态，现有技术必须要求服务处理器（如中央处理单元）介入。服务处理器通过中断或其它机制介入。服务处理器识别多个计时器通道状态，作为回应，在计时器输出管脚产生一个相应的逻辑电平改变。匹配通道状态的一个例子是匹配是否出现。

在识别被选通道状态之后，服务处理器执行状态比较，产生下述结果之一以供特定的自动化应用使用：（1）如果所有被选通道的状态是真则置位输出管脚；（2）如果所有被选通道的状态是真则清零输出管脚；（3）如果任意一个被选通道的状态是真则置位输出管脚；（4）如果任意一个被选通道的状态是真则清零输出管脚。不幸的是，通过服务处理器来比较不同通道的状态和导致输出管脚被驱动到合适的输出逻辑电平需要大量的软件开销。因为置位和清零一个精确定时的输出管脚状态以作为多个通道的综合状态，必须服务处理器的介入，所以服务处理器的等待时间会导致在精确定时的输出中出现定时误差。

本发明允许配置计时器通道以执行上述四类“逻辑”操作，毋需服务处理器的介入。其结果是，在精确定时的输出中不会因服务处理器的等待时间而导致出现定时误差。可以逻辑合并的计时器通道输出的数量理论上仅受限于该计时器系统中可用的通道总数。

本发明提供用户可编程的计时器通道输出管脚逻辑与和或，如图5中的管脚165。在一实施例中，本发明提供了一种系统，该系统拥有多个独立计时器通道和具备选择性配置一部分通道的能力，可以控制管脚状态而毋需

服务处理器的介入。其算法为下述之一：（1）如果所有被选通道的输出状态是真，则置位该管脚；（2）如果所有被选通道的输出状态是真，则清零该管脚；（3）如果任一被选通道的输出状态是真，则置位该管脚；（4）如果任一被选通道的输出状态是真，则清零该管脚。

参看图5，每一个管脚有一个或多个控制寄存器。匹配通道160有控制寄存器166，匹配通道161有控制寄存器167，管脚控制通道有控制寄存器184。图7示出了图5中控制寄存器166的一部分和控制寄存器167的一部分。在一实施例中，控制寄存器166有四个管脚选择控制比特180，用于选择16个集成电路管脚中的某一个，而控制寄存器167有四个管脚选择控制比特181，用于选择16个集成电路管脚中的某一个。在图5示出的本发明的一实施例中，管脚控制选择比特180和181由用户编程以选择管脚165。

在本发明的一实施例中，控制寄存器166有三个管脚输出激励控制比特182，用于选择匹配通道160在输出管脚165产生的功能。同样，控制寄存器167有三个管脚输出激励控制比特183，用于选择匹配通道161在输出管脚165产生的功能。图8示出了在本发明的一实施例中，用户可以通过管脚输出激励控制比特182和183编程的功能。在本发明的可选实施例中，可以使用更多比特，更少比特或不同比特编码不同的控制比特。

参看图8，用户编程功能决定了当匹配发生时，每一个匹配通道160—161怎样影响置位导线157，清零导线158和翻转导线159。例如，如果匹配通道160被编程成屏蔽功能，当匹配发生时，匹配通道160将不对导线157—159产生影响。如果匹配通道160被编程成上升功能，当匹配发生时，匹配通道160将取正置位导线157（通过在置位导线157上驱动逻辑电平0）一个计时器周期。如果匹配通道160被编程成下降功能，当匹配发生时，匹配通道160将取正清零导线158（通过在置位导线158上驱动逻辑电平0）一个计时器周期。如果匹配通道160被编程成翻转功能，当匹配发生时，匹配通道160将取正翻转线159（通过在置位导线159上驱动逻辑电平0）一个计时器周期。

剩余的四项功能，即与清零，与置位，或清零和或置位，是用户可编程的计时器通道输出的逻辑与和或。图9示出了通过匹配通道160和161状态的逻辑与和或在管脚165产生的输出结果。“T”是真，指示发生了匹配，“F”是假，指示匹配未曾发生。如果匹配通道160和161都编程为与置位，

当匹配通道160和匹配通道161都发生匹配时，管脚165的下一个逻辑电平才能被置位（也即是逻辑电平“1”）。这样，与置位意味着匹配必须在匹配通道160和匹配通道161都发生时，才能置位管脚165。同样，与清零意味着匹配必须在匹配通道160和匹配通道161都发生时，才能清零管脚165。

如果匹配通道160和161都编程为或置位，当匹配通道160或匹配通道161发生匹配时，管脚165的下一个逻辑电平被置位（也即是逻辑电平“1”）。这样，或置位意味着在匹配通道160或匹配通道161发生匹配时，就置位管脚165。同样，或清零意味着在匹配通道160或匹配通道161发生匹配时，就清零管脚165。尽管图9示出的例子使用两个匹配通道，实际上可以使用任意数量的匹配通道。

另外，可以合并不同的逻辑与和或功能。例如，附加通道，如计数通道185，可以放置于图5的匹配通道160和管脚控制通道52之间。计数通道185可以在控制寄存器187中有用户编程以选择管脚165的控制管脚选择比特。这样，计数通道185可以驱动导线157—159。如果匹配通道160和161都编程为与置位，计数通道185编程为或置位，当计数通道185达到其计数值或匹配通道160和匹配通道161都发生匹配时，管脚165将被置位。本发明可用于组成更复杂的涉及不同通道状态的布尔方程。

参看图5，因为置位，清零和翻转导线本身的线结异或属性，它们实际上用于执行逻辑与和逻辑或操作。下面描述逻辑与和逻辑或操作是如何执行的。在本发明的一实施例中，置位导线157和清零导线158用于实现与清零和与置位功能；翻转导线159用于实现或清零和或置位功能。

如果匹配通道160编程为与置位，匹配通道160将取正清零导线158直至发生匹配。注意置位导线157和翻转导线159保持预充电的负状态（也即逻辑电平“1”）。当发生匹配时，匹配通道160将取正置位导线157并持续取正直至管脚165的逻辑电平变为逻辑电平1（也就是置位）。注意匹配通道160通过输入状态总线144接收管脚165的输出逻辑电平（参看图4）。只要管脚165保持置位，匹配通道160不再取正任何导线157—159。管脚165可以由其它导线清零，如果匹配通道160被软件重新配置，也可以由匹配通道160对它清零。一旦管脚165被清零，匹配通道160再次取正清零导线158直至发生匹配。

如果匹配通道160编程为与清零，匹配通道160将取正置位导线157直至发生匹配。注意清零导线158和翻转导线159保持预充电的负状态（也即逻辑电平“1”）。当发生匹配时，匹配通道160将取正清零导线158并持续取正直至管脚165的逻辑电平变为逻辑电平1（也就是清零）。注意匹配通道160通过输入状态总线144接收管脚165的输出逻辑电平（参看图4）。只要管脚165保持清零，匹配通道160不再取正任何导线157—159。管脚165可以由其它导线置位，如果匹配通道160被软件重新配置，也可以由匹配通道160对它置位。一旦管脚165被置位，匹配通道160再次取正置位导线157直至发生匹配。

如果匹配通道160编程为或置位，匹配通道160将不取正任何导线157—159直至发生匹配。注意导线157—159保持预充电的负状态（也即逻辑电平“1”）。管脚165必须由其它通道清零，或通过不同配置的匹配通道160清零。当发生匹配时，匹配通道160将取正翻转导线159并持续取正直至管脚165的逻辑电平变为相反的逻辑电平1（也就是置位）。注意匹配通道160通过输入状态总线144接收管脚165的输出逻辑电平（参看图4）。只要管脚165保持置位，匹配通道160不再取正任何导线157—159。管脚165可以由其它导线清零，如果匹配通道160被软件重新配置，也可以由匹配通道160对它清零。一旦管脚165被清零，匹配通道160再次等待直至翻转导线159发生匹配。

如果匹配通道160编程为或清零，匹配通道160将不取正任何导线157—159直至发生匹配。注意导线157—159保持预充电的负状态（也即逻辑电平“1”）。管脚165必须由其它通道置位，或通过不同配置的匹配通道160置位。当发生匹配时，匹配通道160将取正翻转导线159并持续取正直至管脚165的逻辑电平变为相反的逻辑电平1（也就是清零）。注意匹配通道160通过输入状态总线144接收管脚165的输出逻辑电平（参看图4）。只要管脚165保持清零，匹配通道160不再取正任何导线157—159。管脚165可以由其它导线置位，如果匹配通道160被软件重新配置，也可以由匹配通道160对它置位。一旦管脚165被清零，匹配通道160再次等待直至翻转导线159发生匹配。

注意图5中电路操作的上述描述也适用于其它通道57—58和图4中示出的其它输出事件总线导线131—133。

现在描述在图4中示出的输入状态总线143—145的操作。在一实施例中，输入状态总线143，144和145每一个各自包括8个管脚导线146，148和150，以及8个管脚/状态导线147，149和151。这样，每一个管脚/状态总线（如118）有8个导线用于传送管脚信息（如112），和8个可单独编程以传送管脚信息或状态信息的导线（如113）。

在管脚控制通道52中的控制寄存器存储比特192用于有选择地判断每一个管脚/状态导线149是在通道58和一个或多个集成电路管脚195之间传送管脚信息，还是仅在通道58之间传送状态信息。同样，在管脚控制通道51中的控制寄存器存储比特191用于有选择地判断每一个管脚/状态导线151是在通道57和一个或多个集成电路管脚194之间传送管脚信息，还是仅在通道57之间传送状态信息。

管脚/状态导线147，149和151可以用作状态导线，状态导线对管脚没有影响，但可以告知其它通道在某个特定通道上发生了事件（如匹配）。其它导线因而可以基于特定通道上发生的事件，同时调整各自的行为。当用作状态导线时，导线149提供了一种硬件联结通道58中不同通道的方式。同样，当用作状态导线时，导线151提供了一种硬件联结通道57中不同通道的方式。例如，匹配通道161可以通过取正导线149中的一个提供匹配发生的信息。这样，一个或多个通道58中的通道监控该导线以判断通道161中是否发生匹配。

管脚导线148用于将集成电路管脚195的目前逻辑状态传送回通道58，管脚导线150用于将集成电路管脚194的目前逻辑状态传送回通道57。更具体地，管脚导线148中的每一个从管脚向通道58回送该导线所对应的集成电路管脚195中的某一个的目前逻辑电平。同样，管脚导线150中的每一个从管脚向通道57回送该导线所对应的集成电路管脚194中的某一个的目前逻辑电平。

这样，本发明允许多个通道（如图4中通道58）向相同管脚/状态总线导线（如管脚/状态总线导线76）提供信息。其结果是，多于一个通道可以影响集成电路管脚的逻辑电平，（如管脚195中的一个），并且多个通道可以是硬件连接的。尽管上述的实施执行逻辑与和或功能，在本发明的可选实施例中，可以执行任意逻辑操作，包括逻辑异或，异与，排它或和更复杂的布尔功能。

全局通道通信总线

参看图10, 在本发明的一些实施例中, 通过一根全局通道通信总线200在藕合到不同管脚/状态总线216—218的通道间传送信息。全局通道通信总线200的目的是在不同IOCM 25—29 (参看图1) 的通道间传送信息, 以及在一个IOCM中藕合到不同管脚/状态总线 (如图10中的216和217) 的通道间传送信息。另外, 在本发明的一些实施例中, 全局通道通信总线200可以通过集成电路管脚223从外部接收信息, 并向输入/输出集成电路22 (参看图1) 提供信息。注意在本发明的一实施例中, 图1中示出的管脚34可能包括管脚213—215和图10中示出的管脚233的一个或多个。

以现有技术, 本地产生的状态和控制信息通过服务处理器与在一个或多个集成电路上的其它独立电路功能块通信。例如, 许多控制应用使用由通道产生的事件 (如匹配, 捕获, 定时等等) 以触发多个独立电路功能块上的不同处理。这些独立电路功能块在物理上可以分配在不同模块上, 甚至在不同集成电路上。在目前实现中, 要求单个事件产生一个中断, 要求中断服务例程触发在每一个独立电路功能块上所采取的行动。另外, 中断服务例程依次触发每一个行动。

通过使用全局通道通信总线200 (参看图10) 取代中断, 本发明允许向所有通道同时广播状态和控制信息。这样, 多个独立电路功能块内的所有操作都可以同步触发, 从而在整个数据处理系统10内部提供了通信全局信号的一致方式 (参看图1)。

由全局通道通信总线200 (参看图10) 提供的全局广播性能消除了每次发生一个必须与藕合到不同管脚/状态总线的不同通道通信的事件时所需的对服务处理器的中断。与使用服务处理器的现有技术实现方式相比, 因为中断服务例程的速度通常比全局通道通信总线200的广播性能的速度慢一个数量级, 所以全局通道通信总线200也导致了更快的通信和响应时间。

另外, 在软件故障期间, 服务处理器通常是不运行的。因为服务处理器不运行, 中断就不能产生, 许多需要相互通信的I/O功能必须被屏蔽。以前, 如果在软件故障期间要求I/O系统执行一些有限的操作, 需要设计特定

的硬件。因为独立电路功能块相互之间可以通过全局通道通信总线200直接通信，所以本发明的广播性能消除了对这些特定硬件的需要。

本发明的广播性能也简化了修改和增加数据处理系统10（参看图1）所需的设计和实现。在数据处理系统10的许多版本中，一个IOCM 25—29内的通道用于执行一种或有限数量的特定功能，为增加原始应用的带宽或修改应用以执行其它任务而重新配置硬件有些困难。全局通道通信总线200提供的广播性能允许多个独立电路功能块（如IOCM 25—29）相互之间可以通信，从而增加了带宽并允许跨IOCM实现不同功能。

参看图10，全局通道通信总线200用于在耦合到不同管脚/状态总线216—218的通道间传送信息。不同管脚/状态总线216—217可以分布在同一个IOCM（如IOCM27）内，或不同管脚/状态总线217—218可以分布在不同IOCM上（如IOCM 26和IOCM 27）。在本发明的一些实施例中，全局通道通信总线200可以与一个集成电路22上的所有管脚/状态总线耦合。

在可选实施例中，全局通道通信总线200可以仅仅与必须与未耦合在同一根管脚/状态总线上的通道通信的管脚/状态总线耦合。例如，如果通道206中没有通道需要接收或向除了通道206外的任何其它通道提供信息，管脚/状态总线218就不必耦合到全局通道通信总线200上。

参看图1，模块件总线24用于执行C管脚U13对位于外部总线接口23的寄存器和位于IOCM 25—29内的通道的寄存器的读和写操作。模块件总线24也传送系统信息，如中断信息。模块件总线24分别通过总线接口单元36—40与每一个IOCM接口。在本发明的一实施例中，全局通道通信总线200仅仅作为模块件总线24的一部分。

然而，在本发明的可选实施例中，全局通道通信总线200不同于模块件总线24。全局通道通信总线200直接向每一个IOCM 25—29中的管脚控制通道（如201—203）寻找路由。全局通道通信总线200直接与每一个管脚控制通道（如201—203）通信而不是通过总线接口单元36—40间接通信。

在分别位于每一个管脚控制通道201—203中的耦合电路220—222各自耦合到一个或多个控制寄存器224—226。作为例子，控制寄存器224中的控制寄存器存储比特可以由用户编程以选择管脚/状态总线216中的哪一根或哪几根导线耦合到全局通道通信总线200上。这样，用户可以通过控制寄存器229—230和管脚控制电路227—228选择哪个通道耦合到管脚/状态总线

216中的哪一根导线，用户还可以通过控制寄存器224和耦合电路220选择管脚/状态总线216中的哪一根导线耦合到全局通道通信总线200上。

在本发明的一实施例中，全局通道通信总线200的每一个都是由弱下拉设备（未示出）下拉成逻辑电平0的线接或导线。这样，如果工作通道212通过耦合电路222将全局通道通信总线200的第一导线置成逻辑电平1，而工作通道210通过耦合电路221将全局通道通信总线200的同一根第一导线置成逻辑电平1，全局通道通信总线200的第一导线的逻辑电平将是逻辑电平1。因此，对线接或导线而言，逻辑电平1是主要的。可选实施例可以使用线接异或导线。

参看图10，在本发明的一实施例中，管脚/状态总线（如216-218）不是线接或导线或线接异或导线。因此，对接收广播值的通道（如204-206），耦合电路（如220-222）从全局通道通信总线200接收该广播值，然后以合适的值驱动管脚/状态总线的被选导线（如216-218）。类似地，对提供广播值的通道（如204-206），耦合电路（如220-222）从管脚/状态总线的适当导线（如216-218）读取数值，然后以合适的值驱动全局通道通信总线200。

其结果是，本发明允许数据处理系统10（参看图10）的用户编程选择集成电路22中，或不同集成电路（如集成电路12）中哪一个或哪几个通道可以向全局通道通信总线200的导线提供信息。向全局通道通信总线200提供的信息向用户选择的集成电路22或不同集成电路（如集成电路12）的所有通道同步广播，无需服务处理器的介入。

本发明提供了数据处理系统10所有通道的信号或信息的同步全局通信方式。本发明提供了在多个独立电路功能块（如IOCM 25-29）间共享信息的灵活性，从而允许IOCM 25-29共同运行，协调它们执行的一个或多个功能。另外，本发明减少了用于服务处理器的中断数量，允许数据处理系统10可以同步执行更多的功能。

图11示出了图10中控制寄存器226的一部分的一实施例。在本发明的一实施例中，全局通道通信总线200包括8根导线。全局通道通信总线200的8根导线的每一根都有一个对应的SFUN（状态功能控制）寄存器存储比特250，一个SDAT（状态数据）寄存器存储比特251，一个GLS（全局/本地选择控制）寄存器存储比特252，和一个GDO（全局数据输出）寄存器存储

比特253。例如，导线246（参看图12）对应于SFUN比特254，SDAT比特255，GLS比特256和GDO比特257。

SDAT比特251含有状态数据，状态数据代表出于通道到通道的通信和通道到C管脚U的通信的需要，由全局或本地通道产生的输出状态事件的结果。每一个SDAT比特251受对应的SFUN比特250控制。每一个SFUN比特250指定了一种与对应的SDAT比特251相关的模式。SDAT比特251所赋的模式是“硬件控制”模式和“软件控制”模式。除了SFUN比特250，SDAT比特251也由GLS比特252配置。每一个GLS比特252控制相应SDAT比特251是否代表本地状态数据或全局状态数据。

作为例子，参看图10和图12，如果SDAT比特255被相应GLS比特256配置成本地方式，并且被相应SFUN比特254配置成“硬件控制”，SDAT比特255包含由通道206在导线241上驱动的通道输出状态事件的结果。如果SDAT比特255被相应GLS比特256配置成全局方式，并且被相应SFUN比特254配置成“硬件控制”，SDAT比特255包含由集成电路22（参看图1）中任意处的被选通道在导线246上驱动的全局状态事件的结果。SDAT比特255通过导线240向通道206提供它的数据。在“硬件控制”方式下，SDAT比特251不能被C管脚U13写（参看图1）。

如果SDAT比特255被相应GLS比特256配置成本地方式，并且被相应SFUN比特254配置成“软件控制”，SDAT比特255包含由C管脚U 13写入的数据以仿真导线241的状态。然而，当配置成全局方式时，在软件控制方式下，SDAT比特255不能被写入。SDAT比特255反映导线246的全局状态。SDAT比特255向导线240提供该值。在软件控制方式下，仅在配置成本地状态时，SDAT比特251才能被C管脚U13写入（参看图1）。为了写全局状态，GDO比特257必须被写入。

当写SDAT比特251时，可以强制单个SDAT比特变为新状态而不影响其它比特。为实现此目的，SDAT比特251和SFUN比特250必须同时被C管脚U 13写入。因为每一个SFUN比特250控制一个对应的SDAT比特251，所以SFUN比特250的内容决定该次写入影响SDAT比特251。为了写入以影响SDAT比特251中的某一个，相应的SFUN比特250必须写入%0。为了写入而不影响SDAT比特251中的某一个，相应的SFUN比特250必须写入%1。

通过这种方式，SFUN比特250用于屏蔽相应的SDAT比特251以控制对特定SDAT比特251的比特写。

参看图11和图12，GDO比特253包含提供给全局通道通信总线200的全局数据。存储在GDO比特257中的数据由两个可能的源中的一个提供，即导线241和C管脚U 13。SFUN比特250和GLS比特256决定导线241或C管脚U 13是否向GDO比特257提供全局数据。

当GLS比特256被配置成全局方式，GDO比特257用于向导线246提供一个全局状态值。在软件控制方式和硬件控制方式下，GDO比特257都通过在导线246上驱动该全局状态值来提供该全局状态值。如果GDO比特257由相应SFUN比特254配置成软件控制方式，则C管脚U 13可以修改存储在GDO比特257中的全局状态值。如果GDO比特257由相应SFUN比特254配置成硬件控制方式，则只有导线241能修改存储在GDO比特257中的全局状态值。

当GLS比特256被配置成本地方式，GDO比特257不再用于向导线246提供全局状态。GDO比特257不能被C管脚U 13写入。GDO比特257只能被导线241修改。

参看图3，注意将管脚110的一个（对应于PCC 105）藕合到管脚/状态信息导线115的一个（对应于其它成对的PCC 107）的能力，允许用户从全局通道通信总线200向输出管脚110提供通道输出。这样，集成电路22（参看图1）中的任何通道可以通过全局通道通信总线200向任意输出管脚31 - 35输出事件。在本发明的一实施例中，输入管脚110的逻辑电平不在管脚/状态信息导线115上反映，这样，管脚输入不能通过全局通道通信总线进行全局通信。在本发明的一个可选实施例中，一个输入管脚110的逻辑电平可以被反映到管脚/状态信息导线115上，因此可通过全局通道通信总线200来全局传输管脚输入。

在本发明的一个实施例中，GDO比特位253能被CPU13随时读取，仅在软件控制模式中设置为全局状态时，才可被CPU13写入。

参照图11和图12，每个SFUN比特位250用于选择模式，称为软件控制模式或硬件控制模式，分别对应一个SDAT比特位251和GDO比特位253。CPU13可以随时读写SFUN比特位250。

参照图11和图12，每个GLS比特位252用于选择对应的一个SDAT比特位251是局部的或是全局的。存储在SDAT比特位251的局部状态值

仅在局部管脚/状态总线(如图 10 中的 218)上传输,该总线由局部管脚控制通道(如图 10 中的 203)所控制。存储在 GDO 比特位 257 中的全局状态值,可以通过全局通道通信总线 200 或局部管脚/状态总线(如图 10 中的 218)来提供。CPU13 可以随时读写 GLS 比特位 252。

参照图 12,应注意 SDAT 比特位 255 可由对应的 GLS 比特位 256 分别设置为全局的或局部的。设置为全局时,导线 241 提供一个全局状态值给 GDO 比特位 257,而不是 SDAT 比特位 255。GDO 比特位 257 是通过导线 246 向全局通道通信总线 200 提供全局状态值的来源。引线或导线 246 的最终逻辑电平(即全局状态值)存储在 SDAT 比特位 255 中。该全局状态值传输到导线 240,再送至通道 206 的输入(见图 10)。

应注意本发明的一个实施例中,有一个全局计时器总线(未标出),耦合给每个主计时器总线控制通道(如图 2 中的 61 和 62),用于将一个或多个时基值传输给不同的集成电路 22(如图 1)上的计时器总线(如图 2 中的 71 和 72)。全局计时器总线(未示出)以模拟方式作用在全局通道通信总线 200 上(见图 10)。在本发明的一个可选例中,未使用一个全局计时器总线。而是,如果独立的 IOCMs25 - 29 要求相同的时基值时,产生所需时基的时基通道(如图 2 中的度计时器通道 94)被复制给多个 IOCMs25 - 29。

通道间数据传输

在“通道硅库(silicon library of channels)”多个通道,可以编程用于完成数据传输操作。在数据传输操作中,数据可以从一个顶端相邻通道(如图 20 中的 400)的数据寄存器传输给通道本身的数据寄存器(如 401),和从通道(如 401)本身的数据寄存器传输给底端相邻通道(如 402)的数据寄存器。通过对通道的控制寄存器比特位进行编程,来完成这些通道间数据传输,就可以构成和使用栈和先入先出(FIFO)结构。在本发明的一个实施例中支持通道间数据传输的工作通道包括:(1)匹配通道(2)俘获通道;和(3)加计数器通道。在一个实施例中,可逆计数器通道也允许一些数据传输功能。

在本发明的一个实施例中,可利用支持通道间数据传输的通道构成三种基本类型的结构:(1)栈,是收集许多时基值或计数操作值的有用结

构；（2）FIFO，多个匹配输出事件可以输出给一个管脚或状态导线，其中利用存储在 FIFO 结构中所收集的时基值，（这些输出事件可被其它通道或直至 I/O 集成电路等装置所利用）；（3）FIFO 也可用于收集和存储最新俘获到的时基值或最新计数操作。

这种栈和 FIFO 数据存储结构可以降低通道要求的服务频率，因而降低需由 CPU13 响应的中断数（见图 1）。图 20 显示相邻通道 400 - 402 间数据传输操作的一个例子。

如图 20 所示，匹配通道 400，俘获通道 401，计数器通道 402，各自控制从它们顶端相邻通道到它们自身数据寄存器的数据传输。为了控制数据传输，一个通道的数据传输逻辑（如 407 - 409 中的一个），与顶端相邻通道的数据传输逻辑进行通信，每个通道的数据传输逻辑可以向状态总线 414 产生两种输出文件，称为显示通道与顶端相邻通道都有有效数据的一种输出文件，和显示通道与顶端相邻通道都有无效数据的一种输出文件。此外，俘获通道 401 可以利用来自状态总线 414 的输入文件，使它数据寄存器 404 中的数据无效。

在本发明的一个实施例中，管脚控制通道（PCC）（如图 2 中的 52）具有一个 32-bit 数据通路，经过它，PCC 的顶端相邻通道可以穿过 PCC 传输数据给它的底端相邻通道。穿过 PCC 的数据传输不影响 PCC。（即 PCC 没有数据传输控制和不存储通过它的数据。）

图 21 和图 22 所示电路，包括控制和状态寄存器存储电路，它被用来支持一个通道（如俘获通道 401）中的数据传输操作。图 21 所示电路用于 16-bit 数据传输操作，图 22 所示电路用于 32-bit 数据传输操作。32-bit 数据传输功能，用于数据传输操作的寄存器比特位，相邻通道间通信的信号，所用的输入事件和一个通道数据传输逻辑产生的输出事件都将在以下描述。应注意：控制和状态寄存器存储电路可被看作虚拟的盒子，因为它们物理上定位于一个或多个用户可编程寄存器的一部分。应注意：清零 DVB 输入事件信号仅能被俘获通道使用。

匹配通道 400，俘获通道 401，计数器通道 402 都可被设置运行在数据传输模式，或者作为两个完全独立的 16 位子通道或者作为一个 32 位通道。如图 21 所示，俘获通道 401 被设置成 16 位数据传输操作模式。如图 22 所示，俘获通道 400 被设置成 32 位数据传输操作模式。匹配通道 400

和计数器通道 402 可用同样的方式设置。

参照图 21，应注意：控制每个 16 位子通道的逻辑单元功能相似但彼此独立。在本发明的一个实施例中，一个通道的两个 16 位子通道不必运行于同一操作模式。例如：设置匹配通道 400 的高 16 位子通道，来匹配计时器总线上的 16 位值(如图 2 的计时器总线 71)，低 16 位子通道用于 16 比特数据传输操作。

参考图 21，控制通道高 16 位的寄存器存储比特位被置“0”，控制通道低 16 位的寄存器存储比特位被置“1”（如 DTC0 和 DTC1）。如图 22，当通道（如 401）被设置进行 32 比特数据传输时，控制高 16 位子通道的寄存器存储位被用来控制 32 位的数据传输。

在本发明的一个实施例中，支持数据传输操作的通道有三种不同的寄存器比特字段，用来控制数据传输和向 CPU13 提供状态信息。这三个寄存器位分别是数据有效比特（DVB），数据传输控制比特（DTC），和数据传输状态比特（DTS）。在本发明的一个实施例中，俘获通道和计数器通道还使用输入事件边沿选择位（IE）来控制事件传输操作。

参考图 21 和 22，每个数据有效比特位（DVB）425 - 426 被相应的数据传输逻辑 422，429，430 用作控制和状态比特位。作为状态比特位，DVB 指示相应数据寄存器中有效或无效数据的出现。作为控制比特位，DVB 被相应的数据传输逻辑用来控制从顶端相邻通道的数据传输。

在本发明的一个实施例中，DVB 位为逻辑“1”，表示该通道数据寄存器中出现有效数据。例如，在俘获通道 401 中，只要俘获操作一完成，DVB 位就被置为逻辑“1”。如图 20 和图 22 所示，当数据传入数据寄存器 404（如从顶端相邻通道 400），俘获通道的数据传输逻辑 430 就置 DVB 位为逻辑“1”。

此外，如果 CPU13（见图 1）向任何被设为数据传输模式的数据通道的数据寄存器写入时，DVB 位被自动置逻辑“1”，显示该数据有效。在本发明的一个实施例中，当匹配通道处于匹配模式，向其数据寄存器写入时，匹配通道的 DVB 位亦被置逻辑“1”。

在本发明的一个实施例中，DVB 位逻辑“0”表示通道数据寄存器中出现无效数据。例如，在匹配通道 400 中，一旦匹配输出事件发生，DVB 位清为逻辑“0”。如图 20 和图 22 所示，当数据从数据寄存器 404 传入

数据寄存器 405（底端相邻通道），传输逻辑 409 清零俘获通道 401 中的 DVB 位 425 为逻辑“0”。一旦数据被传输，通道 401 的 DVB 比特 425 被清零为逻辑“0”，显示数据不再有效。

此外，CPU13 能通过清零相应 DVB 位 425 为逻辑“0”，使数据寄存器 404 中的数据无效。为了清零 DVB 位 425，CPU13 必须先确立态读取 DVB 位 425，然后必须向其写入逻辑“1”。DVB 位 425 为逻辑“1”显示数据寄存器 404 中的数据为无效。应注意：由于 DVB 位既为状态位又为控制位，CPU13 对 DVB 位的清零动作可能会引起有效数据被数据传输操作覆盖。

对本发明的一个实施例，图 23 列出能置位和清零 DVB 位的操作。包括在各种操作模式下，对每种支持数据传输操作的通道的操作。参考图 20 和图 22，当通道 402 被设置为数据传输模式，顶端相邻通道的 DVB 位（通道 401 的 DVB 位）为逻辑“1”，通道本身的 DVB 位（如通道 402 的 DVB 位）为逻辑“0”时，发生数据传输。通道的数据传输逻辑（如通道 402 的传输逻辑），首先从顶端相邻的通道数据寄存器 404，拷贝数据到该通道的数据寄存器 405。接着顶端相邻通道的 DVB 位（通道 401 的 DVB 比特位）被清零为逻辑“0”。最后该通道的 DVB 位被置为逻辑“1”。

参考图 21 和图 22，数据传输控制（DTC）比特位 423 - 424 被用来使能和禁止相应的数据传输逻辑 422，429，430。当一个通道的数据传输逻辑被使能，顶端相邻通道的数据寄存器的内容可被传输至该通道的数据寄存器。当一个通道的 DVB 位为逻辑“0”，且顶端相邻通道的 DVB 位为逻辑“1”时，数据传输开始。

在数据传输操作的下列两种模式下，产生输出事件。（1）如果通道和其顶端相邻通道的 DVB 位皆为逻辑“1”（表明两通道都出现有效数据），开始传输数据并引起一个输出事件。（这个输出事件被称为有效相邻数据对（VADP）输出事件）（2）如果通道和其顶端相邻通道的 DVB 比特皆为逻辑“0”（表明两通道都出现无效数据），开始传输数据并引起一个输出事件。（这个输出事件被称为无效相邻数据对（IADP）输出事件）

在本发明的一个实施例中，这两种模式下输出事件的目的（输出事件导线）和输出事件的类型（上升，下降，翻转）都不由软件控制。这两种

输出事件都与同一状态总线 414（见图 21，22）直接相连，且二者都引起一个触发事件。

图 24 描述了由可进行数据传输操作的通道中的数据传输控制位（DTC）（如图 21，22 中的 423 - 424）控制的数据传输操作模式。

参照图 21，22，数据输出状态位（DTS）427 - 428 被用来标志一个输出事件已被相应数据传输逻辑 422，429，430 产生。如果 DTC = % 10，且 VADP 检测输出事件已由相应的事件传输逻辑产生，则 DTS 位被置为逻辑“1”。如果 DTC = % 11，且 IADP 检测输出事件已由相应的事件传输逻辑产生，则 DTS 位被置为逻辑“0”。为了清零 DTS 比特 427，CPU13 必须先确立态读 DTS 位 427，然后向其写入逻辑“1”。

下面将讨论用来实现数据传输操作的控制信号。参照图 21，22，图中标有大量的数据传输控制信号，这些信号用来与（或来自）设置在数据传输操作模式的顶端相邻通道通信。两通道联络的三种途径：32/16 - BIT 数据线，读信号线，清零信号线。32/16 - BIT 数据线用于通道的数据寄存器（如图 20 中通道 401 的数据寄存器 404）从顶端相邻通道（如图 20 的通道 400）接收数据。

读信号用于通道的数据传输逻辑，与顶端相邻通道的 DVB 位之间的通信。读取的信息有两个用途。第一，在数据传输操作时，如果通道本身的 DVB 为逻辑“0”且顶端相邻通道的 DVB 为逻辑“1”，那么开始传输数据。第二，当通道被编程产生一个 VADP 或一个 IADP 输出事件，读信号用来决定是否在一个有效或无效的相邻数据对。

清零信号被用于顶端相邻通道，在数据从该顶端相邻通道的数据寄存器，传至通道本身的数据寄存器之后，有条件地清零通道的 DVB 比特。

应注意：图 22 中数据传输通道有三种途径与底端相邻通道联络。它们是已经提到的：数据，读，写信号线。在设计集成电路时，如果让数据传输通道（如图 20 的 400 - 402）电路彼此邻接，则数据，读，写信号线的布线可直接从一个通道，指向相邻的另一个通道。在本发明的一个实施例中，可以让每个数据传输通道，控制从相邻通道至本身数据寄存器的数据传输。在本发明的可选实施例中，使用不同的状态和控制寄存器比特，不同的数据传输控制信号，在 I/O 集成电路 22 中采用不同的电路在相邻通道间传输数据（见图 1）。

在本发明的一个实施例中，所有数据传输通道都产生两种类型的输出事件：一个有效相邻数据对输出事件（VADP），一个无效相邻数据输出事件（IADP）。此外，在本发明的一些实施例中，运行于数据传输模式的俘获和计数器通道，可以利用一个输入事件清零俘获或计数器通道本身的 DVB 比特（见图 21，22）。

参考图 21，22，在本发明的一个实施例中，有效相邻数据对(VADP)，或无效相邻数据对（IADP）输出事件信号，是输出至状态线 414 并产生一个触发事件。VADP 和 IADP 输出事件，可被 CPU13（见图 1）用来显示一个堆栈满或一个 FIFO（先入先出队列）空。管脚控制通道（PCC）对状态线 414（如管脚/控制总线 76 的状态线）进行监控，它可被编程，在状态线上的触发事件被识别时引起中断。

俘获通道或计数器通道的输入事件逻辑可被编程，在进行数据传输操作时，使用一个输入事件，清零相应的 DVB 比特（见图 21，22 中的清零 DVB 输入事件信号 431 和 433）。这对禁止一个 FIFO 队列中的底层数据传输通道的数据极为有用，这样位于底层数据通道之上的所有数据都可以沿一个通道向下传输。

在计时器应用中有三种基本的数据传输结构。第一种数据传输结构是堆栈，它保存俘获的时基值和计数操作值。第二种数据传输结构是时基值的 FIFO 队列，用它来产生一系列匹配输出事件。第三种数据传输结构是保存大量最近俘获的时基值，和最新计数操作值的 FIFO 队列。本发明容许用户编程设置这三种基本时基传输结构，来调整数据传输能力和一个或多个通道的操作。

计数操作控制特征

许多基于控制应用的微控制器，需要对目标信号进行高精度的累加测量。例如：图 2 中的计数器通道 58 可对目标信号进行高精度的累加测量。目标信号可由外部提供进入输入/输出（I/O）集成电路 22（见图 1），也可由内部提供送入输入/输出（I/O）集成电路 22。计数器通道 58（见图 2）可进行多种计数操作，并通过向它的一个或多个用户可编程控制寄存器存储比特 67，写入预置值来选择操作方式。

如图 26 所示，作为第一个可编程选择的操作，计数器通道 58 可根据

目标信号电平的高和低决定增加或减少计数值。第二，计数器通道 58 可在每次目标信号的有效沿到达时增加计数值。有效沿可编程设定为目标信号的上升沿或下降沿。第三，计数器通道可在有效沿首次到达时开始连续计数。第四，计数器通道 58 可在每次目标信号的有效沿到达时减少计数值。本发明的可选实施例可使用别的计数操作。

一些控制应用需要由第二个通道（如匹配通道 56，俘获通道 55，通道组 87 中的别的计数器通道）产生的第二路信号，决定何时进行目标信号的计数操作。在现有技术中，如果计数器通道在一时间段内对目标信号进行累加操作，第二路信号应对目标信号进行选通，或者向中断服务处理器产生一个中断。中断服务处理器将响应中断，选通计数器通道对目标信号的计数操作。

同样在现有技术中，为了停止计数器通道对目标信号的计数，目标信号选通的第二路信号应变为无效，或者向中断服务处理器产生一个中断。中断服务处理器将响应中断，中止计数器通道对目标信号的计数操作。不幸的是，采用现有技术时常导致计数操作的累加错误。

如图 25 所示：在现有技术中，当用第二路来中止计数器通道对目标信号的计数操作，就引入了累加错误。因为第二路信号在目标信号下降沿到来之前已变为无效，即计数器通道在目标信号的一个周期结束之前就停止计数了。这样存储在现有技术计数器通道的周期累加值会低于正确值，其差值恰好为累加错误的值。应注意：图 25，26 中向上的箭头指示计数器通道 58 作增数操作，向下的箭头指示计数器通道 58 作减数操作。

参考图 25，用现有技术在一时间段内进行周期累加，如在“计数窗口”内，”计数窗口“由计数窗口信号确立有效开始，反转为无效结束。通常由第一个计数器通道对目标信号事件计数，由第二个通道产生计数窗口信号。第一个计数器通道可接收目标信号和接收窗口信号。如图 25 所示，计数窗口信号的确立开始状态为“高”电平，目标信号的有效沿为下降沿。

参考图 25 所示的采用现有技术进行周期累加，当计数窗口信号确立开始，第一个计数器通道在目标信号紧接的有效沿开始计数，并在计数窗口信号有效时不断计数。当第一个计数器通道接收到的计数窗口信号变为无效，第一个计数器通道立刻停止计数。这样，如果计数窗口信号没有目标信号的周期边界上变为无效（如图 25 所示），就会导致累加错误。

然而，本发明允许计数器通道接收第三个通道（如图 2 中，匹配通道 56，俘获通道 55，通道组 87 中别的计数器通道等）提供的第三路信号，或者连至 I/O 集成电路 22(见图 1) 的外部源提供的第三路信号。图 26 中标记为“计数停止信号”的第三路信号，被计数器通道 58 接收来决定何时停止接收操作。应注意：对计数器通道 58 的使能和禁止不需处理器的任何中断服务。此外，使用第三路信号来中止计数操作，允许计数操作停止在目标信号的周期边界，从而避免了累加错误。

作为计数停止信号的第三路信号，提供了在一时间段内，准确无误累加目标信号事件的途径。应注意：在现有技术中，没有办法利用由外部或其它通道提供的第三路信号，来增强中止计数操作。此外，利用第三路信号提供了不需中断服务响应，便能准确无误测量一时间段内目标信号的累加值的有效途径。

应注意：在本发明的一些实施例中，目标信号亦可用作计数停止信号。参考图 25，计数停止条件可编程设置为目标信号的某一边沿类型。例如：通过向计数器通道 58（见图 2）的控制寄存器存储位 67，写入预置值来选择计数停止信号为目标信号的下降沿。这样当计数窗口信号反转为无效，且计数停止事件发生时(如目标信号的下降沿)，计数器通道 58 停止计数。这样，计数器通道 58 在计数窗口信号确立开始后，目标信号的第一个下降沿开始计数，在计数窗口信号反转为无效后目标信号的第一个下降沿停止计数。结果，计数器通道 58 测量并保存了非常精确的周期累加值，现有技术无法与之相比。

本发明允许利用第二路，第三路信号来优化对目标信号的计数操作：计数操作由第二路信号定义的“计数窗口”选通，由第二路或第二路和用来限制第二路的第三路信号来中止。在某些情况下，目标信号的边沿可用作第三路信号。在一个实施例中，计数器通道 58（见图 2）的控制寄存器存储位 67 的某些位，被用来决定是否使用计数停止信号。而且有些位用来选择中止计数操作的事件，是第三路信号（图 26 中称为计数停止信号）事件，还是图 25 所示的目标信号事件，还是现有技术的计数窗口信号事件。在现有技术中，计数窗口信号的反转总是用于中止计数操作。

图 27 给出了计数器通道 58（见图 2）一部分的一个实施例。计数器通道包括一个计数器电路 440 和一个数据寄存器 447。数据寄存器与总线

24（见图1）直接耦合。计数器电路440包括一个计数器441，一个控制电路442，另一个控制电路443。控制电路442从线路448接收接收窗口信号，从线路449接收计数停止信号，从线路450接收目标信号。控制电路442通过线路444，向计数器提供计数使能信号。控制电路443通过线路445，向计数器提供计数禁止信号。计数器441还从线路450上接收目标信号。计数器441通过线路446向数据寄存器447提供计数值。

控制和状态信息通过导线451在计数器441，控制电路442，控制电路443之间传输。例如，控制电路442和443中的一个通过导线451，提供一个数据传输控制信号给计数器441。当提供给计数器441的数据传输控制信号确立后，计数器441将现在的计数值通过导线446传输给数据寄存器447。数据寄存器447可通过总线24利用读写存取来访问。计数器441通过导线452接受一个计时器信号。控制寄存器存储比特位67通过导线453耦合给控制电路442和443。

图28显示了寄存器67一部分的一个实施例（见图2和图27）。应注意按照计数器通道58所选模式，一些寄存器比特位具有不同的功能。例如，当计数窗口模式无效时，CZO/WTO/DTO比特位被用作计数到零输出比特位，当计数窗口模式有效时，被用作窗口终端输出比特位，当选择数据传输模式时，被用作数据传输输出比特位。在一个实施例中，图28所示控制寄存器比特位执行下列功能。

16/8 和 24/32 - 16 - 比特，8 - 和 24 - 比特，或 32 - 比特操作比特位 469

00 - 32 - 比特操作

01 - 组合 8 - 和 24 - 比特操作

1X - 16 - 比特操作

CCS：计数器计时器选择比特位 470

0 - 计数器利用输入事件计时

1 - 计数器利用计时器总线输入计时

DTC：数据传输控制比特位 471

0X - 计数模式有效

10 – w/VADP 检测输出事件有效

11 – w/IADP 检测输出事件有效

CLK： 计时器总线源选择比特位 472

000 – 111 – 计时器总线连线 0 – 7

SCC： 单个/连续计数操作比特位 473

0 – 连续计数操作

1 – 单个计数操作

CI： 计数输入比特位

0000 – 0111 – 管脚连线 0 – 管脚连线 7

1000 – 1111 – 状态连线 0 – 状态连线 7

CIEL： 计数输入边沿或电平比特位 475

CCS=0

X00 – 输入无效

X01 – 计数上升沿输入事件

X10 – 计数下降沿输入事件

X11 – 两边沿计数输入事件

CCS=1

000 – 输入无效

001 – 利用计时器在上升沿开始

010 – 利用计时器在下降沿开始

011 – 利用计时器在任一沿开始

100 – 利用计时器在输入低开始

101 – 利用计时器在输入高开始

110 – 利用计时器当输入低时计数

111 – 利用计时器当输入高时计数

USI/MTI/CDVI： 加计数停止输入/

模传输输入/

清零数据有效输入比特位 476

0000 – 0111 – 管脚 0 – 管脚 7

1000 – 1111 – 状态 0 – 状态 7

USIE/MTIE/CDVE：加计数停止输入边沿/

模传输输入边沿/

清零数据有效边沿比特位 477

00 – 输入无效

01 – 在上升沿停止 cnt/Xfer mod/Clr DVB

10 – 在下降沿停止 cnt/Xfer mod/Clr DVB

11 – 在任一沿停止 cnt/Xfer mod/Clr DVB

CZO/WTO/DTO：计数到零输出/

窗口终端输出/

数据传输输出比特位 478

0000 – 0111 – 管脚连线 0 – 管脚连线 7

1000 – 1111 – 状态连线 0 – 状态连线 7

CZOEL/WTOEL/DTOE：计数到零输出边沿或逻辑操作/

窗口终端输出边沿或逻辑操作/

数据传输输出边沿比特位 479

CZOEL/WTOEL

000 – 输出无效

001 – 输出一个上升沿

010 – 输出一个下降沿

011 – 输出一个翻转

100 – 条件与输出置位管脚/状态

101 – 条件或输出置位管脚/状态

110 – 条件与输出清零管脚/状态

111 – 条件或输出清零管脚/状态

DTOE

X00 – 输出无效

X01 – 输出一个上升沿

X10 – 输出一个下降沿

X11 – 输出一个翻转

图 29 示出控制寄存器 67（见图 27）一部分的一个实施例。在一个实施例中，图 29 所示控制寄存器比特位执行下列功能。

CM：计数器模式比特位 480

0 – 减计数模式

1 – 加计数模式

CWE：计数窗口使能比特位 481

00 – 计数窗口操作无效

01 – 利用所选状态连线使能

1X – 利用来自低端相邻双 FIFO 通道（如可用）的输入使能

应注意：如果一个低端相邻双 FIFO 通道不存在并选择 **CWE=1X**，则窗口输入总是低。

WTOC：窗口终端输出控制比特位 482

00 – 计数到零输出有效，

窗口终端输出无效

01 – 计数到零输出无效，

窗口终端输出对一个窗口内第一个计数操作有效

10 – 计数到零输出有效，

窗口终端输出对一个窗口内第一个计数操作有效

11 – 计数到零输出有效，

窗口终端输出对一个窗口内任意计数操作有效

OFC：溢出控制比特位 483

0 – 溢出时停止在零

1 – 溢出时继续增加

LCE：逻辑补使能比特位 485

0 – 逻辑补输出无效

1 – 逻辑补输出有效

CWI：计数窗口输入比特位 484

000 – 111 – 状态连线 0 – 状态连线 7

图 30 显示寄存器 67（见图 27）一部分的一个实施例。应注意在某些实施例中，寄存器 67 包括控制寄存器比特位和状态寄存器比特位。在图 30 所示实施例中，寄存器 67 的一部分用来存储状态信息。在一个实施例中，图 30 所示状态寄存器比特位执行下列功能。

WTS：窗口终端状态比特位 490

0 – 窗口内一个单个计数操作，或第一个连续计数操作没有被终止

1 – 窗口内一个单个计数操作，或第一个连续计数操作被终止

USIS/MTS/CDVS：加计数停止输入状态/模传输状态/清零数据有效状态比特位 491

0 – 停止计数器/模传输/清零数据有效没有发生

1 – 停止计数器/模传输/清零数据有效发生

CZS/DTS：计数到零状态/数据传输状态比特位 492

0 – Inc/Dec 到零或 VADP/IADP 没有发生

1 – Inc/Dec 到零或 VADP/IADP 发生

DVB：数据有效比特位 493

0 – 数据寄存器内数据无效

1 – 数据寄存器内数据有效

CIS：计数输入状态比特位 494

0 – 加/减计数输入没有发生

1 – 加/减计数输入发生

图 31 显示计数器 441（见图 27）中的计数操作，与图 28 – 29 所示一些控制寄存器比特位之间的关系。参照图 31，当 $CCS = 0$ 和 $CIEL = X01$ 时，计数窗口信号为高后，计数器 441 在每个上升边沿输入时增加。应注意在计数窗口信号变为低后，计数器 441 继续增加。计数窗口信号变为低后，当第一个计数停止信号事件产生后，计数器 441 停止。在这一点，计数操作完成，控制寄存器 67 中的加计数停止输入状态比特位被置位，计数值被传输给数据寄存器 447，控制寄存器 67 中的 DVB 比特位被置位。

当 $CCS = 1$ 和 $CIEL = 111$ 时，无论何时目标信号是高和计数窗口信号时高，计数器 441 利用 CLK 比特字段所选的计时器输入进行加计数。此外，计数窗口信号变为低后，当第一个计数停止信号事件产生时，计数器 441 停止。当 $CCS = 1$ 和 $CIEL = 010$ ，计数器 441 利用来自导线 452 的计时器信号进行加计数，当计数窗口信号为高时，在目标信号的第一个下降边沿开始计数操作。同样，当计数窗口信号反转后，计数器 441 在第一个计数停止信号事件产生时停止。

图 32 显示了计数器通道 58（见图 2）的一部分的一个可选实施例。在这个实施例中，计数器通道 58 包括一个计数器寄存器 460，一个事件寄存器 461，和输出事件逻辑 467。在一个实施例中，计数器寄存器 460 可被分成能独立操作的两个计数器部分。第一个计数器部分响应计数器控制逻辑 462 和输入事件逻辑 464。第二个计数器部分响应计数器控制逻辑 463 和输入事件逻辑 465。在一个实施例中，每个计数器部分也有一套独立的，如图 28 和图 29 所示的控制寄存器比特位。

图 32 所示各种控制和状态比特位，尽管位于可存取读写寄存器内，也影响如图 32 所示计数器通道 58 的行为。应注意 DTC 寄存器比特位 471，DVB 寄存器比特位 493，和 DTS 寄存器比特位 492，以如图 20 – 24 中和本说明的附带部分所描述的同样方式，用于通道间数据传输。

应注意在一个实施例中，如果计数器（如图 27 中的计数器 441）是一个减计数器时，则不需要计数停止信号。在使用了一个减计数器的一些实施例中，一个全零的计数值可以用来指示应停止计数操作。这样对于一个减计数器，一个全零的计数值可以执行与加计数器中的一个计数停止信号同样的功能。这样，在计数窗口信号反转后，当计数值第一次达到全零时，一个减计数器停止计数。

俘获窗口特征

许多基于微控制器的控制应用要求一个时基值，该时基值仅在一个目标信号的上升沿，下降沿，或在两者处俘获。例如，基于所选一个目标信号的边沿产生的时间，利用图 2 中的俘获通道 55 从计时器总线 71 俘获时基值。目标信号可从外到输入/输出（I/O）集成电路 22（见图 1）提供，或产生给内至 I/O 集成电路 22。俘获通道 55 可执行各种俘获操作，该操作可由预先写入俘获通道 55（见图 2，5，35 和 36）中，一个或更多个用户可编程控制寄存器存储比特位 66 的控制值来选择。

许多控制应用要求一个由一个第二通道产生的第二信号，用于基于一个目标信号产生的时间，来规定何时俘获一个时基值和何时不俘获一个时基值。例如，参照图 2，这个产生第二信号的第二通道可以是通道 54，它可以是一个匹配通道，一个计数器通道，或通道 57 中另一个其它类型的工作通道。理想情况下，时基值俘获的使能和禁止不应由一个处理器（如图 1 中的 CPU13）要求直接服务，并且俘获时基值应与目标信号的频率一致无关。

然而，在现有技术控制应用中，要求第二信号产生一个中断，要求引入一个处理器（如图 1 中的 CPU13）执行的中断服务例程，用于在俘获通道内使能俘获操作。此外，为在俘获通道内禁止俘获操作，要求第二信号产生一个中断，要求引入中断服务例程，用于在俘获通道内禁止俘获操作。

另外，为保持俘获时基值的一致，现有技术的控制应用一般要求中断服务处理，用于在下一个目标信号的触发沿之前，将每个俘获时基值传输给一个锁存器或存储器。这样如果目标信号的频率足够高时，中断服务例程并不总是能够在下一个目标信号的触发沿产生之前，将俘获时基值传输给一个锁存器或存储器，因而失去俘获时基值的一致性。

图 33 显示出一个第二信号，即所谓一个俘获窗口信号，是如何用于规定从一个计时器总线俘获时基值。作为一个第一可编程选择，俘获通道 55 可以基于目标信号的上次事件（即上次触发沿）俘获和存储一个时基值。触发沿可以编程选择为目标信号的上升沿，下降沿或任意沿。第二，俘获通道 55 与“ $N - 1$ ”其他俘获通道一起，可以俘获和存储“ N ”时基值，每个时基值对应一个“ N ”上一次目标信号触发沿。第三，俘获通道 55 与“ $N - 1$ ”其他俘获通道一起，可以俘获和存储“ N ”时基值，每个时基值对应一个“ N ”上一次目标信号正边沿。另外，俘获通道 55 与“ $N - 1$ ”其他俘获通道一起，可以俘获和存储“ N ”时基值，每个时基值对应一个“ N ”上一次目标信号负边沿。第四，俘获通道 55 与“ $N - 1$ ”其他俘获通道一起，可以俘获和存储“ N ”时基值，每个时基值对应一个“ N ”第一次目标信号触发沿。在本发明的一个实施例中，使用了两个俘获通道，则数目“ N ”是 2。本段中使用的“ N ”代表一个正整数。本发明的可选实施例可以使用其他俘获操作。

如上所述，俘获窗口特征消除了一个处理器（如图 1 中的 CPU13）服务介入的需要，并提供一个数据相关，而与目标信号频率无关。俘获窗口特征允许一个俘获通道直接接收激励（即俘获窗口信号），来决定何时俘获时基，何时不俘获。因此，一个处理器（如图 1 中的 CPU13）的中断服务被大大减少，并保持俘获时基值的一致，并与目标信号的频率无关。

俘获窗口特征允许一个俘获通道对时基值的俘获，由一个第二信号的逻辑状态来规定，该信号称为一个俘获窗口信号，这样时基值的俘获可以在第二信号定义的窗口期间被使能。

一个俘获窗口使能寄存器比特位（图 35 中的 CWE501）被用于使能俘获窗口特征。一个俘获窗口是一个时段或时间/角度的“窗口”，其中执行时基俘获操作。当俘获窗口比特位被置位时，基于目标信号边沿的时基俘获由一个第二信号的状态来规定，该信号称为俘获窗口信号。属于目标信号的时基俘获，当俘获窗口信号状态确立时（图 33 - 34 中的逻辑电平 1）被允许，当第二信号状态反转时（图 33 - 34 中的逻辑电平 0）被禁止。

图 34 显示使用俘获窗口特征，和使用位于俘获通道 55（见图 2）的寄存器 66 的某些寄存器控制比特位的一个例子。在这个例子中，俘获窗口使能比特位（图 35 中的 CWE501）被置位，因而使能俘获窗口特征。俘

获通道 55 在目标信号的每个上升沿产生时,俘获在计时器总线 71 (见图 2) 上的时基值。应注意, 尽管如此, 俘获通道 55 也仅在俘获窗口信号确立和所选目标信号边沿产生时, 俘获在计时器总线 71 上的时基值。因此, 俘获窗口信号被确立的时段定义了允许俘获的“窗口”。当俘获窗口信号反转时, 进一步的目标信号时基俘获被禁止。

应注意图 34 所示例子中, 俘获窗口信号通过管脚/状态总线 75, 或从底端相邻双 FIFO 通道 (如被实现) 被提供给俘获通道 55。在一个实施例中, 一个双 FIFO 通道是一个工作通道, 能够作为一个双重深度 (two-deep) FIFO 用于存储数值。例如, 参照图 20, 计数器通道 402 可以被一个双 FIFO 通道所取代, 这样作为底端相邻双 FIFO 通道用于俘获通道 401。如果 CWE 比特位 501 选择管脚/状态总线 75, 作为俘获窗口信号的来源, 则利用 CWI 比特位 505, 来选择管脚/状态总线导线 75 中的哪一个被用于向俘获通道 55 提供俘获窗口信号。如果 CWE 比特位 501 选择底端相邻双 FIFO 通道作为俘获窗口信号的来源, 则 CWI 比特位 505 被忽略。应注意, 如果底端相邻通道不是一个双 FIFO 通道, 而且 CWE 比特位 505 被编程选择一个底端相邻双 FIFO 通道, 则俘获窗口保持无效。

仍参照图 34 所示例子, 利用寄存器 66 (见图 35) 中的 CI 比特位 506, 来选择管脚/状态总线导线 75 中的哪一个被用于向俘获通道 55 提供俘获目标信号。利用 CIE 比特位 507 来选择俘获目标信号的哪一个边沿, 所谓上升沿, 下降沿, 或任一边沿, 被用于触发从计时器总线 71 俘获时基值。而且, 利用 CTB 比特位 503, 来选择俘获通道 55 从计时器总线 71 俘获哪一个时基值。

图 35 显示了寄存器 66 (见图 2 和图 27) 的一部分的一个实施例。应注意基于为俘获通道 55 所选的模式, 一些寄存器比特位可以有一个不同的功能。例如, 当俘获窗口模式被使能时, CI/CDVI 比特位 506 用作俘获输入比特位, 当选择数据传输模式时, 用作清零数据有效输入比特位。在一个实施例中, 图 35 所示控制寄存器比特位执行下列功能。

16/32: 16 - 比特或 32 - 比特功能比特位 500

0 - 32 - 比特

1 - 16 - 比特

CWE：俘获窗口使能比特位 501

00 – 俘获窗口无效

01 – 利用 CWI 比特字段所选状态线使能俘获窗口

1x – 利用从底端相邻双 FIFO 通道（如果可用）的输入使能俘获窗口

DTC：数据传输控制比特位 502

0x – 数据传输无效；俘获有效

10 – VADP

11 – IADP

CTB：俘获时基比特位 503

000 – 111 时基 0 – 7

SCC：单个/连续俘获比特位 504

0 – 连续俘获操作

1 – 单个俘获操作

CWI：俘获窗口输入比特位 505

000 – 111 状态总线 0 – 7

CI/CDVI：俘获输入/清零数据有效输入比特位 506

0000 – 0111 管脚总线 0 – 7

1000 – 1111 状态总线 0 – 7

CIE/CDVE：俘获输入边沿/清零数据有效边沿比特位 507

00 – 输入无效

01 – 上升沿

10 – 下降沿

11 – 任一沿

CO/DTO：俘获输出/数据传输输出比特位 508

000 – 111 状态线 0 – 7

COE/DTOE：俘获输出边沿/数据传输输出边沿比特位 509

00 – 输出无效

01 – 上升沿

10 – 下降沿

11 – 翻转

图 36 显示寄存器 66（见图 2）的一部分的一个实施例。应注意在一些实施例中，寄存器 66 包括控制寄存器比特位和状态寄存器比特位两者。在图 36 所示实施例中，寄存器 66 的一部分用于存储状态信息。在一个实施例中，图 36 所示状态寄存器比特位执行下列功能。

CES：俘获边沿状态比特位 510

0 – 检测到一个下降沿输入事件，产生相关的俘获操作

1 – 检测到一个上升沿输入事件，产生相关的俘获操作

CDVS：清零数据有效状态比特位 511

0 – 没有检测到 DVB = 1 的规定输入事件

1 – 检测到 DVB = 1 的规定输入事件

COS：俘获输出状态比特位 512

0 – 俘获操作没有产生或 COE 比特字段 = 00

1 – 俘获操作产生和 COE 比特字段 = 00

DTS：数据传输输出状态比特位 512

0 – 没有检测到符合 VADP 或 IADP 的条件

1 – 检测到符合 VADP 或 IADP 的条件

DVB：数据有效比特位 513

- 0 - 数据寄存器中的数据无效
- 1 - 数据寄存器中的数据有效

CIS: 俘获输入状态/清零数据有效状态比特位 514

- 0 - 没有产生一个俘获
- 1 - 产生一个俘获

图 37 显示俘获通道 55 (见图 2) 的一部分的一个实施例。在这个实施例中, 俘获通道 55 包括一个数据寄存器 520, 俘获逻辑 521, 数据传输逻辑 522, 单个/连续俘获逻辑 523, 输出事件逻辑 524, 输入事件逻辑 525, 和 16/32 比特逻辑 526。

图 37 所示各种控制和状态比特位, 尽管位于可存取读写寄存器内, 也影响如图 37 所示俘获通道 55 的行为。应注意 DTC 寄存器比特位 502, DVB 寄存器比特位 513, 和 DTS 寄存器比特位 512, 以如图 20 - 24 中和本说明的附带部分所描述的同样方式, 用于通道间数据传输。

一个单个/连续操作计时器通道的特征

在 I/O 集成电路 22 (见图 1) 的一个实施例中, 每个工作通道都有一个单个操作模式和一个连续传输模式。在本发明的一个实施例中, 每个计数器通道 (如计数器通道 58), 每个俘获通道 (如俘获通道 55), 和每个匹配通道 (如匹配通道 57) 都有一个用户可编程寄存器比特位, 允许以单个模式或连续模式操作通道。

例如, 单个/连续计数操作 (SCC) 寄存器比特位 473 (见图 28), 决定计数器通道 58 (见图 2) 是在单个操作模式还是在连续操作模式。单个/连续俘获操作 (SCC) 寄存器比特位 504 (见图 35), 决定俘获通道 55 (见图 2) 是在单个操作模式还是在连续操作模式。而且, 单个/连续匹配操作 (SCM) 寄存器比特位 531 (见图 38), 决定匹配通道 57 (见图 2) 是在单个操作模式还是在连续操作模式。

在进入单个操作模式时, 相关的计时器功能状态比特位 (如图 30 中的计数输入状态 (CIS) 比特位 494, 图 36 中的俘获输出状态 (COS) 比特位 512, 和图 39 中的匹配状态 (MS) 比特位 534) 将被清零, 而提供

计时器功能的状态和控制。当工作通道为单个操作模式设置时，如果相关计时器功能状态比特位被置位，通道将被禁止，如果相关计时器功能状态比特位被清零，通道被使能。

不幸的是尽管如此，对一个单个操作模式/连续操作模式工作通道，仍有三个问题。第一个问题是从一个连续操作模式到一个单个操作模式的一致转换。第二个问题是在单个操作模式时，防止一个匹配通道意外的再使能。第三个问题是在连续操作模式中，防止基于从一个匹配通道的冗余匹配事件的多重输出。

从一个连续操作模式到一个单个操作模式一致转换的问题以下列方式解决。当在单个操作模式（即“单个命中”匹配，俘获等）执行计时器功能时，与计时器功能相关的状态比特位，（即图 30 中的 CIS 比特位 494，图 36 中的 COS 比特位 512，和图 39 中的 MS 比特位 534），显示计时器功能是否有效还是无效。在一个实施例中，如果状态比特位被清零，计时器功能被使能，如果状态比特位被置位，计时器功能被禁止。结果是状态比特位“控制”计时器功能的操作。

当在一个连续操作模式（即“连续”匹配，俘获等）执行计时器功能时，与计时器功能相关的同一个状态比特位显示一个或更多事件（即匹配，俘获等）是否产生。状态比特位仅提供操作信息，而不以任何方式影响或控制计时器功能。

当从连续操作模式转换到单个操作模式时，同样与计时器功能相关的状态比特位，必须从执行一个“状态”功能转换到一个“状态和控制”功能。非常类似于当工作通道是在连续操作模式时，状态比特位将已被置位。当转换到单个操作模式时，如果状态比特位保持置位，工作通道将立即禁止自己，而不用已执行一个计时器功能。如果工作通道在进入单个操作模式时变为无效，可能通道将丢失“激励”，因而导致从连续到单个操作模式的一个不一致的转换。

工作通道电路使得相关的状态比特位，在进入单个操作模式时清零自己，然后允许该状态比特位执行一个“状态和控制”功能。工作通道电路在连续和单个操作模式中都使用同样的状态比特位，尽管使用状态比特位的方法按照模式而不同。另外，当在两模式间转换时，对工作通道的激励将不会丢失，在这一点上两模式间的转换是一致的。

在单个操作模式时，防止一个匹配通道意外的再使能的问题以下列方式解决。当一个匹配通道（如图 2 和图 40 中的匹配通道 57）工作在单个操作模式时（即仅允许产生一个真比较，然后禁止匹配通道），在相关匹配状态比特位（如图 39 中的 MS 比特位 534）显示一个匹配没有产生时（ $MS = \% 0$ ），匹配通道将被使能。当相关匹配状态比特位显示一个匹配产生时（ $MS = \% 1$ ），匹配通道将被禁止。

在某些情况中，需要清零匹配状态比特位并且不再使能匹配通道。匹配通道电路通过仅清零匹配状态比特位，来防止匹配通道被再使能。为了再使能匹配通道，匹配状态比特位必须被清零，并且数据寄存器（如图 40 中的数据寄存器 540）必须被写入。这样提供了一个一致的方式，用于 CPU13（见图 1）控制匹配通道（即改变控制寄存器中存储的控制值，改变匹配寄存器（如图 40 中的 540）中存储的数据值，和反转匹配状态比特位），而不用再使能匹配通道。这是一个有用的特征，允许 CPU13（见图 1）清零匹配状态比特位，而不用再使能匹配功能。

防止当在连续操作模式时，基于冗余匹配事件的多重输出的问题以下列方式解决。在连续操作模式，当一个匹配通道对一个“慢”时基（即以低于匹配速率一半的频率增加）进行匹配时，可能对于同一个时基值有多重比较。当时基值等于数据寄存器中的值并在多重比较中保持相同时，数据寄存器将在每次执行一个比较时，确立一个真比较。

需要在时基值和匹配寄存器中的值进行匹配时，仅确立一个输出，而不管产生了多少“真比较”信号。匹配通道电路仅允许下列真比较信号产生一个输出：（1）从上次假比较后第一个真比较；和（2）从匹配寄存器在它的数据寄存器接收到新数据（即被 CPU13 写入或从一个相邻通道传输进来）后，第一个真比较。当第一个真匹配产生时，一个连续匹配相关被置位（禁止匹配），在对相同时基，上次真比较后第一个真匹配产生时，或在连续模式（即 $SCM = \% 0$ 和 $MOD = \% 000$ ）匹配被禁止时，该相关被清零（使能匹配）。结果是将不产生冗余匹配，而与匹配时基频率无关。

图 38 所示寄存器 65（见图 2）的一部分的一个实施例。在一个实施例中，图 38 所示控制寄存器比特位执行下列功能。

12/16/32：12/半字/字选择比特位 529

0x – 32 – 比特 (字)

10 – 12 – 比特

11 – 16 – 比特

MTB：匹配时基比特位 530

000 – 111 时基 0 – 7

SCM：单个/连续匹配比特位 531

0 – 连续匹配操作

1 – 单个匹配操作

MO：匹配输出比特位 532

0000 – 0111 管脚总线 0 – 7

1000 – 1111 状态总线 0 – 7

MOE：匹配输出边沿比特位 533

000 – 输出和相等比较被禁止

001 – 输出一个上升沿

010 – 输出一个下降沿

011 – 输出一个翻转

100 – 条件与输出置位管脚/状态

101 – 条件或输出置位管脚/状态

110 – 条件与输出清零管脚/状态

111 – 条件或输出清零管脚/状态

图 39 显示寄存器 65（见图 27）的一部分的一个实施例。应注意在一些实施例中，寄存器 65 包括控制寄存器比特位和状态寄存器比特位。在图 39 所示实施例中，寄存器 65 的一部分被用于存储状态信息。在一个实施例中，图 39 所示状态寄存器比特位执行下列功能。

MS：匹配状态

0 - 没有产生一个匹配

1 - 产生一个匹配事件

图 40 显示匹配通道 57（见图 2）的一部分的一个实施例。在这个实施例中，匹配通道 57 包括一个数据寄存器 540，相等比较器逻辑 541，12/16/32 选择逻辑 542，匹配逻辑 543，和输出事件逻辑 544。在一个实施例中，数据寄存器 540 可分为两部分，称为一个上部分和一个下部分，两者可以独立操作。在一个实施例中，每个匹配部分都有独立的一套如图 38 和图 39 所示的控制和状态寄存器比特位，当仅有一套控制划分的 12/16/32 选择比特位 529 时除外。第一匹配部分使用数据寄存器 540，相等比较器逻辑 541，匹配逻辑 543，和输出事件逻辑 544 的第一部分；第二匹配部分使用数据寄存器 540，相等比较器逻辑 541，匹配逻辑 543，和输出事件逻辑 544 的第二部分。

图 40 所示各种控制和状态比特位，尽管位于可存取读写寄存器内，也影响如图 40 所示匹配通道 57 的行为。应注意在图 40 所示实施例中，匹配通道 57 不包括被用来执行相邻通道间数据传输的寄存器比特位和电路。然而，参照图 40 和图 37，在本发明的可选实施例中，匹配通道 57 可以包括数据传输逻辑和寄存器比特位 DTC，DVB，和 DTS，用于执行与俘获通道 55 中数据传输逻辑 522 和寄存器比特位 DTC471，DVB493 和 DTS492 同样的功能。如果实现，将以如图 20 - 24 中和本说明的附带部分所描述的同样方式，产生来自或去往一个匹配通道的数据传输。

仍参照图 40，匹配通道 57 支持单个和连续匹配操作。当以单个或连续匹配操作运行时，数据寄存器 540 中的值与来自计时器总线 71 的八个值中的一个相比较。输出事件可被编程为被产生和提供给管脚/状态总线 76。匹配通道 57 的一个实施例的操作，现在将根据操作目的，设置操作的控制比特位，和产生的输出事件来加以描述。

在单个和连续匹配操作中，匹配产生在有效数据，和来自计时器总线 71 的一个值（任一个 32 - 比特或一个 12/16 比特时基值）之间。时基值由对 MTB 寄存器比特位 530（见图 38）编程来选择。时基值比特位的数目由对 12/16/32 选择比特位 529 编程来选择。例如，如果选择 12 - 比特操作，

时基值的低 12 比特位和数据寄存器 540 中数值的低 12 比特位，将由相同比较器 541 进行比较。

如果 MOE 比特位不等于 % 000，和数据寄存器 540 中的值不等于来自计时器总线 71 上的时基值，这时产生一个匹配，MS 比特位 34（见图 39）被置位。产生一个输出事件，并提供给管脚/状态总线导线 76 中的一个。输出事件的种类由对 MOE 比特位 533（见图 38）编程来选择。如果 MOE 比特位等于 % 000，输出事件逻辑 544 和相等比较器 541 被禁止。输出事件的目的由对 MO 比特位 532（见图 38）编程来选择。

单个/连续匹配（SCM）比特位 531（见图 38）用于在 CPU13（见图 1）软件服务间，选择匹配通道仅产生一个输出事件（即单个操作模式），还是产生多于一个输出事件（即连续操作模式）。

当单个/连续匹配（SCM）比特位 531（见图 38）是 % 1 时，选择单个匹配操作。在单个匹配操作，当来自计时器总线 71 的时基值，和存储在数据寄存器 540 的数值相等时，在管脚/状态总线 76 上产生一个输出事件，并且确立单个匹配相关机制来防止冗余匹配。后续输出事件仅在解除单个匹配相关机制后才能产生。需要下面两个操作来解除单个匹配相关机制（可以任意次序执行这两个操作）：（1）匹配状态（MS）比特位 534（见图 39）在确立状态（即一个逻辑电平 1）被读取，而反转后的状态（即逻辑电平 0）被写回 MS 比特位 534；和（2）一个数值被写入数据寄存器 540。

应注意 CPU13（见图 1）被用来执行单个匹配相关机制要求的读写存取。另外，由于以下原因，需要操作（2）来解除单个匹配相关机制。如果存在产生输出事件的多重匹配通道，和在相同管脚/状态总线导线 76 上的中断请求，操作（2）允许对任意匹配通道清零 MS 比特位 534，这样清零中断源的状态而无需再使能通道。

当单个/连续匹配（SCM）比特位 531（见图 38）是 % 0 时，选择连续匹配操作。在连续匹配操作，当来自计时器总线 71 的时基值，和存储在数据寄存器 540 的数值相等时，在管脚/状态总线 76 上产生一个输出事件，并且确立连续匹配相关机制来防止冗余匹配。后续输出事件仅在解除连续匹配相关机制后才能产生。在一个不相等比较产生或 MOE 比特位 533 被

清零为% 000 后，才解除连续匹配相关。在第一个匹配产生后， MS 比特位 534 被置位，但不需为产生后续匹配清零 MS 比特位 534。

本发明参照特定的实施例进行图解和描述，本领域的技术人员将会想到进一步的修改和改进。因此应该理解，本发明不仅限于图示的特定形式，而且如在附加的权利要求中所表明的，覆盖所有修改后的形式，它们并未超出本发明的精神和范围。

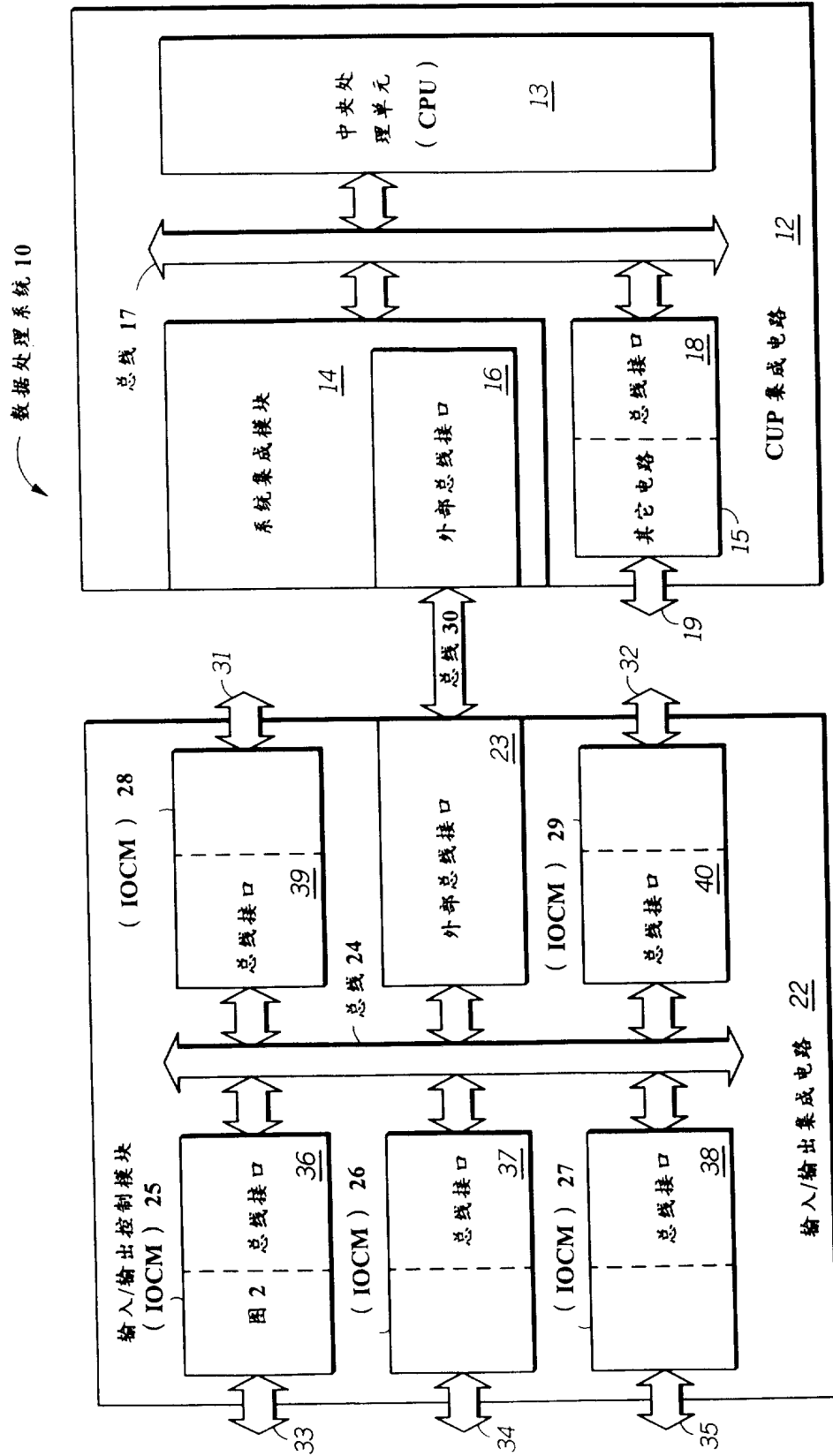


图 1

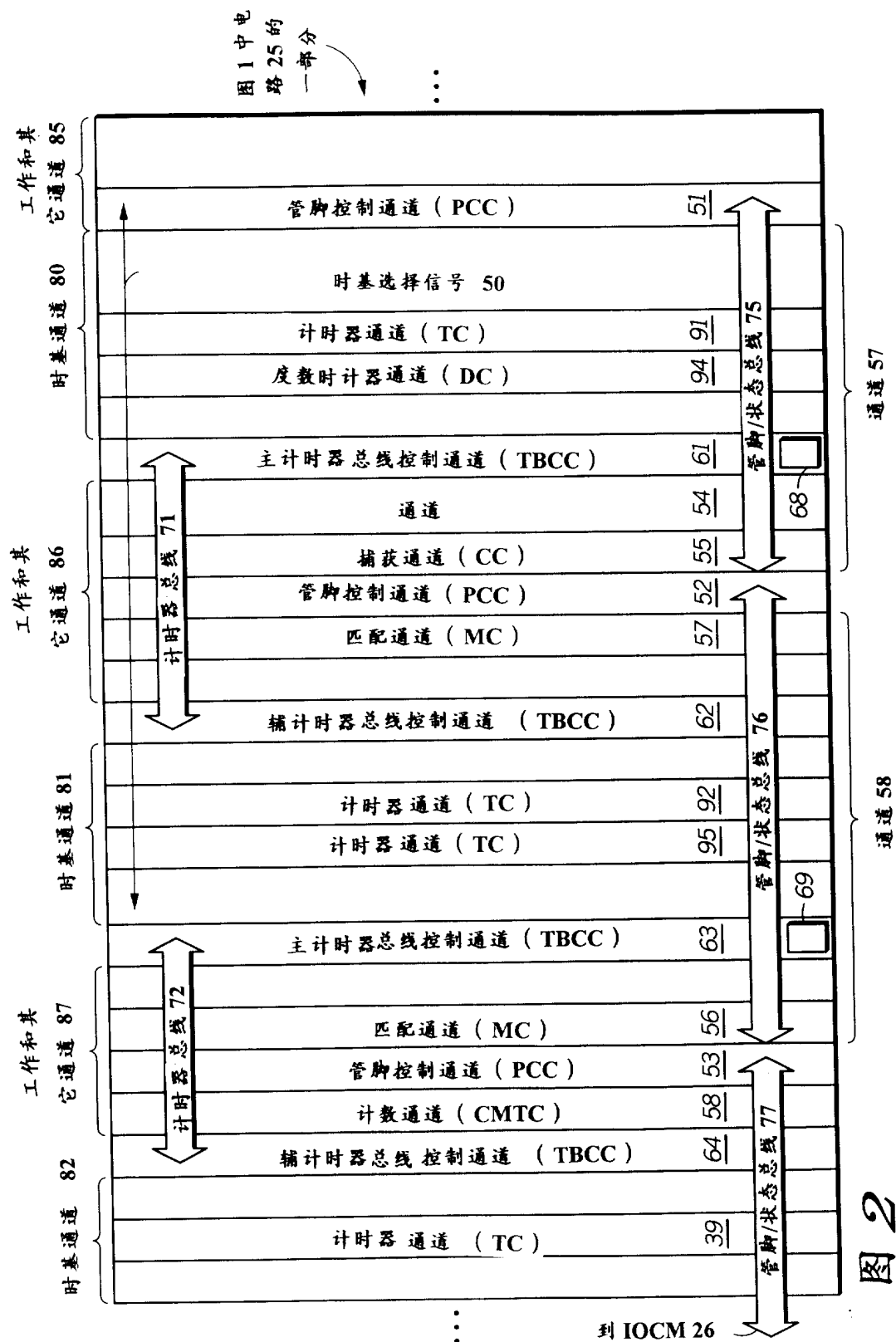


图 2

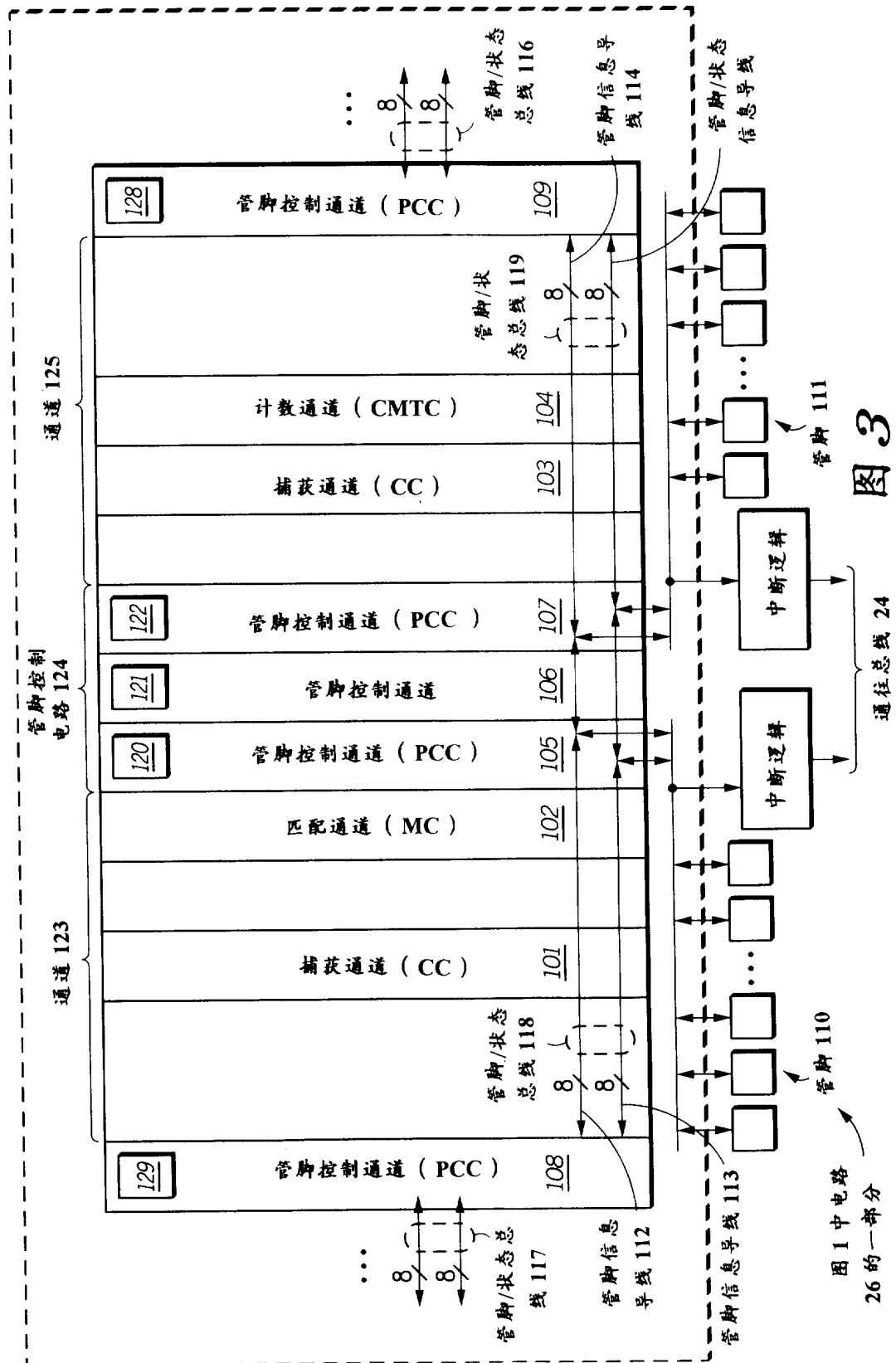


图 1 中电路
26 的一部分

图 3

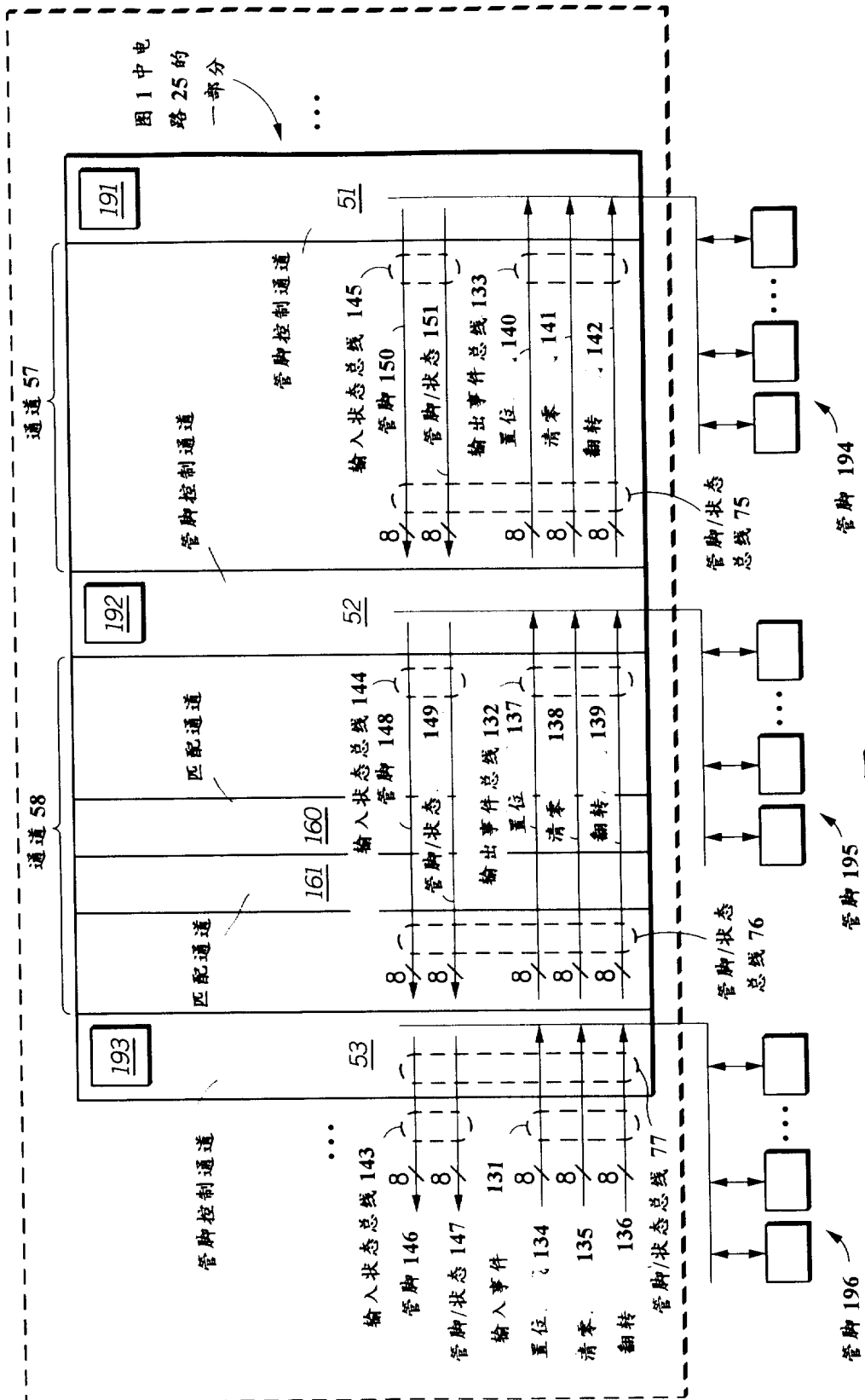


图 4

图 1 中电
路 25 的
一部分

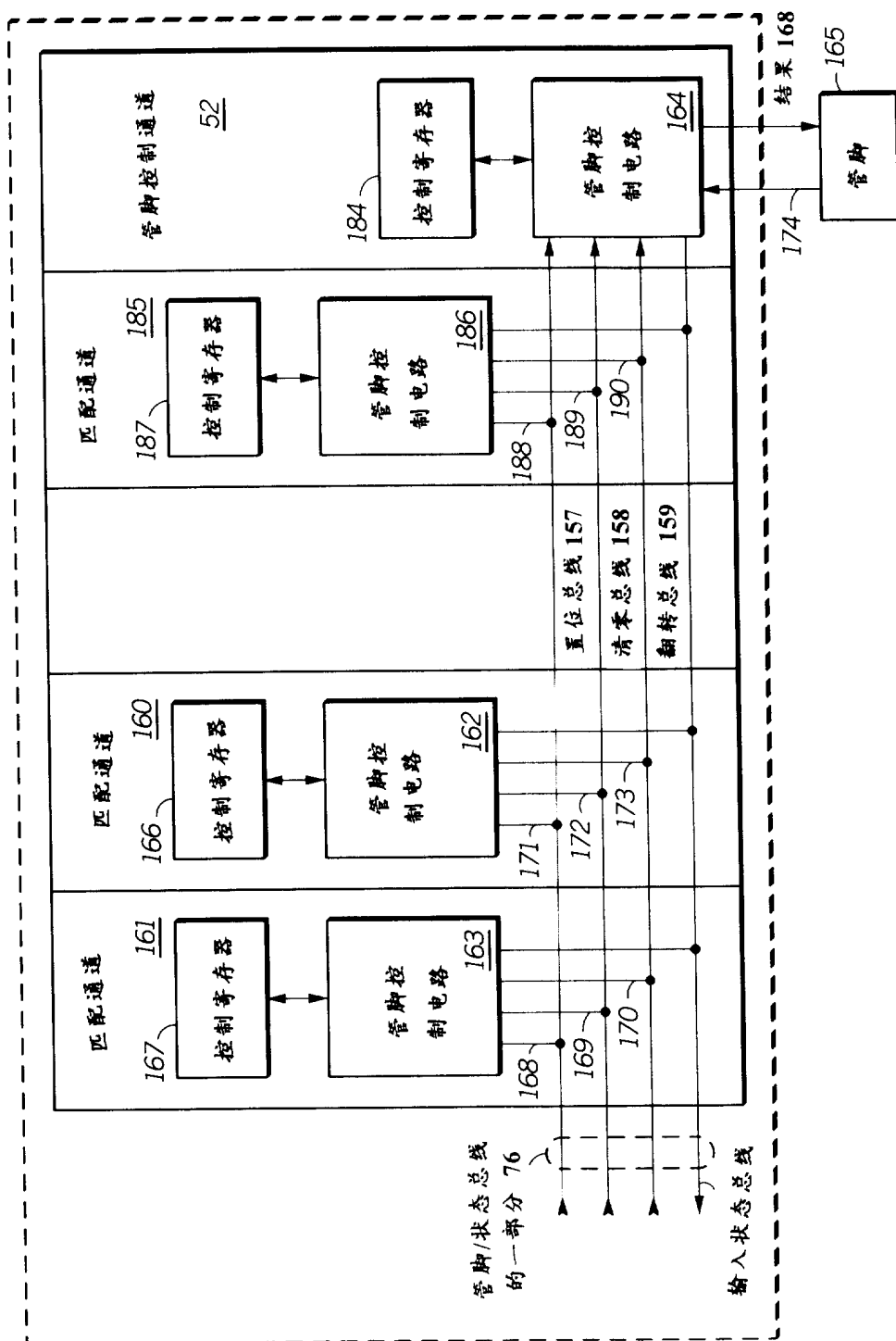


图 5

翻转导线 159 的逻辑电平	置位导线 157 的逻辑电平	清零导线 158 的逻辑电平	结果导线 168 的当前逻辑电平	结果导线 168 的下一个逻辑电平
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

注意：
 逻辑电平“0”是导线 157 - 159 的正状态。
 逻辑电平“1”是导线 157 - 159 的负状态。

图 6

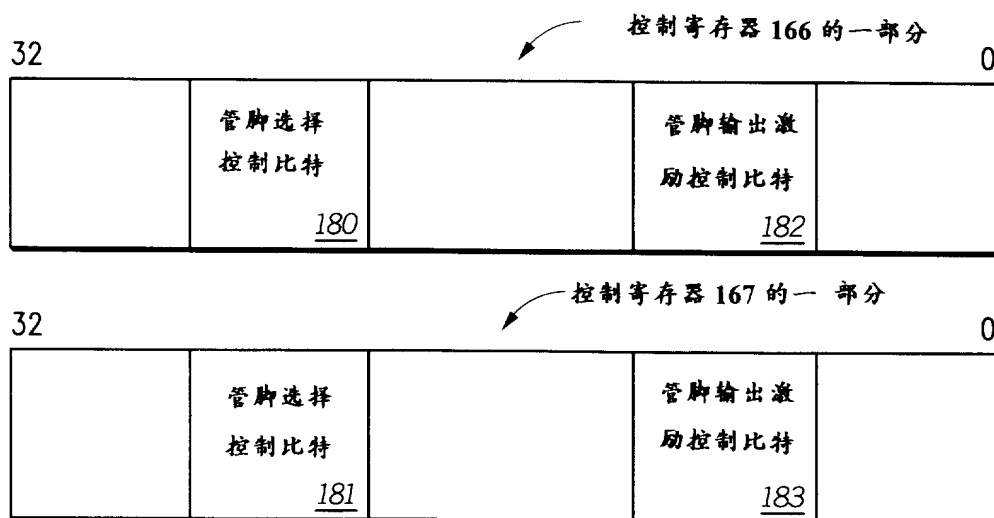


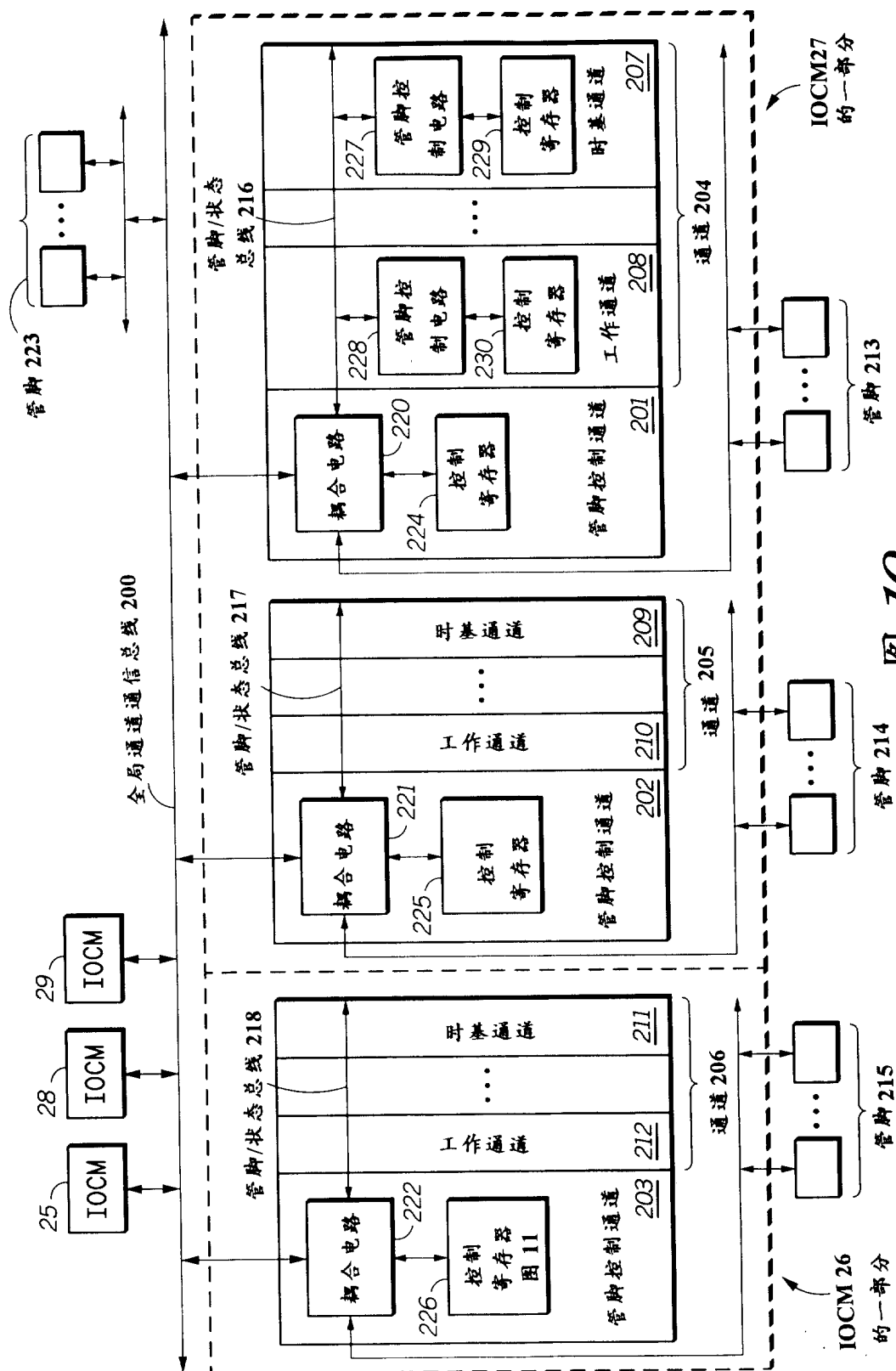
图 7

管脚输出激励控制比物 (182 或 183)	功能
000	屏蔽
001	上升
010	下降
011	翻转
100	与清零
101	与置位
110	或清零
111	或置位

图 8

通道输出的本地组合 (参见图 5)									
匹配通道的输出 160	匹配通道的输出 161	管脚 165 的逻辑电平							
		与置位		或置位		与清零		或清零	
		当前	下一个	当前	下一个	当前	下一个	当前	下一个
T	F	0	0	0	0	1	1	1	1
F	T	0	0	0	1	1	1	1	0
T	F	0	0	0	1	1	1	1	0
T	T	0	1	0	1	1	0	1	0

图 9



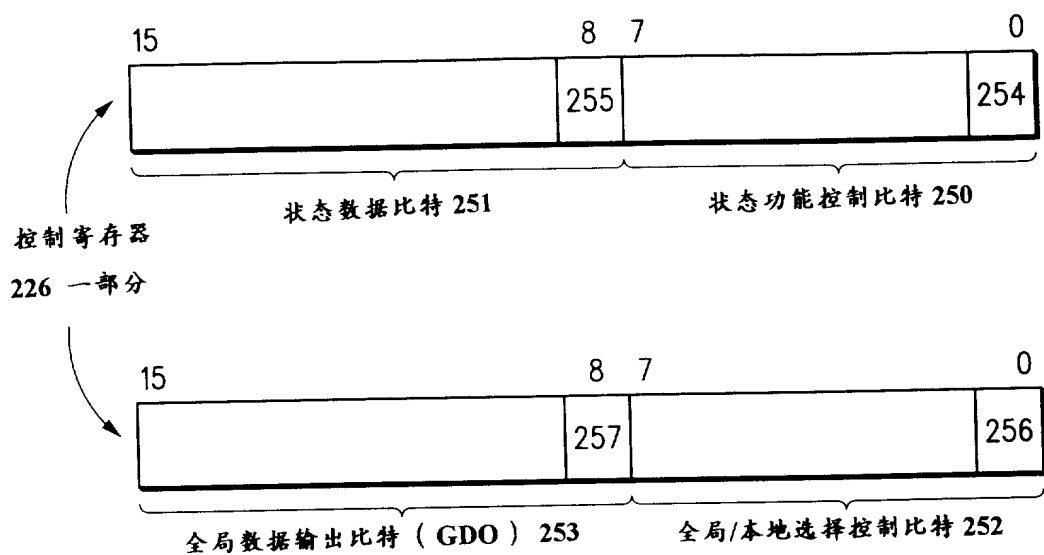


图 11

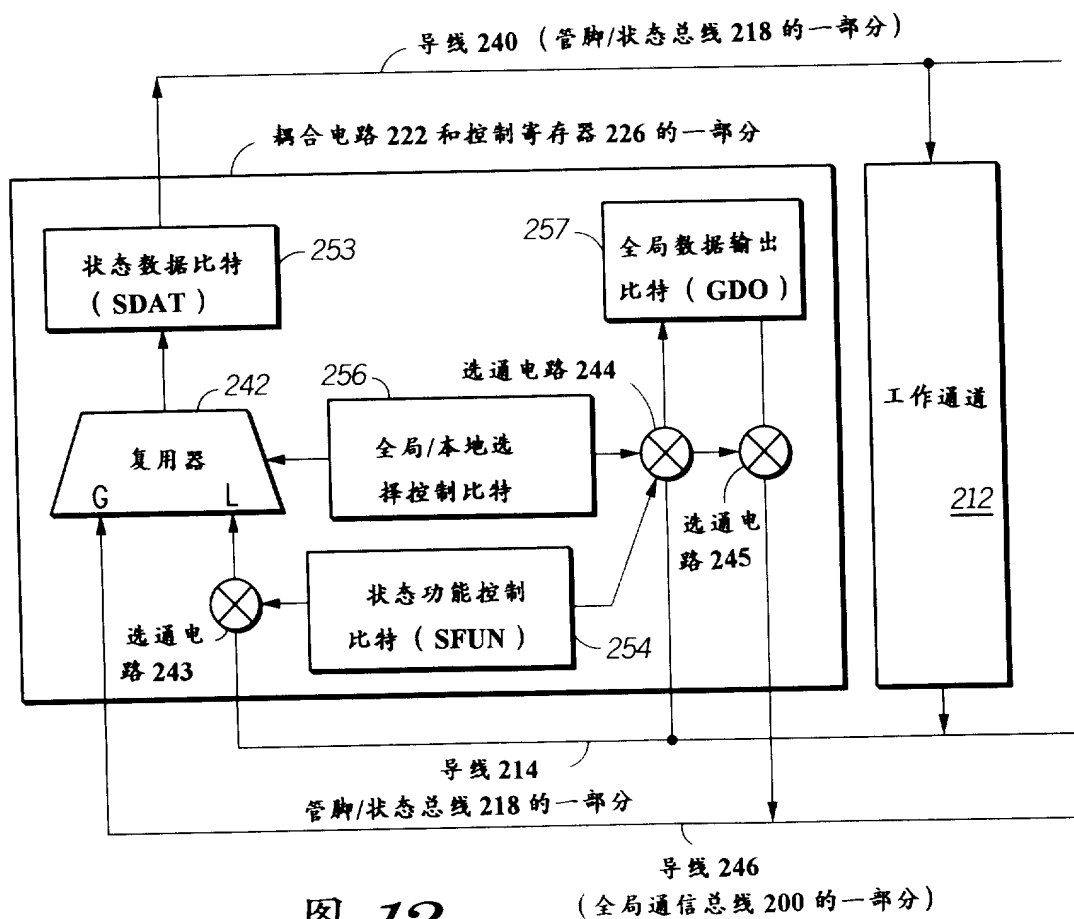


图 12

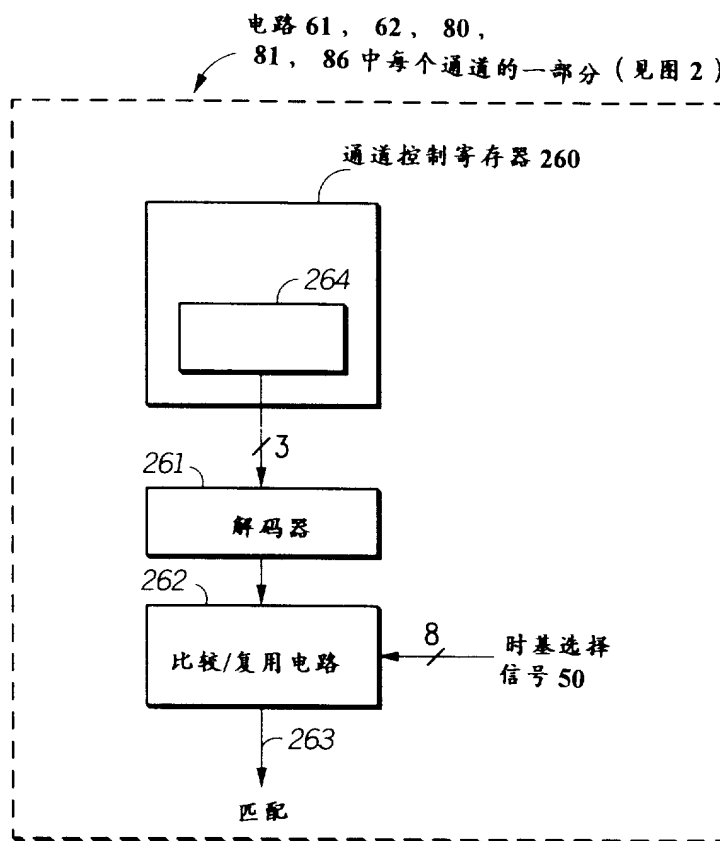


图 13

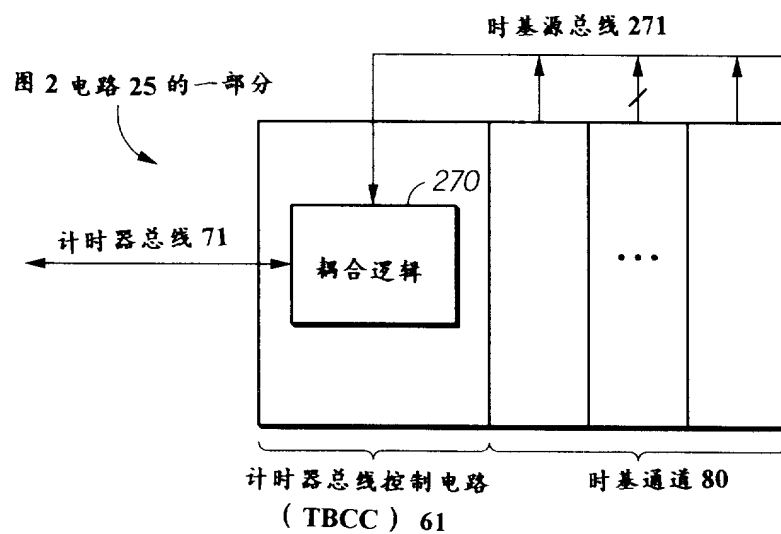


图 14

利用一个时钟总线的 8 个时隙提供时基值的 8 个例子									最大分辨率 (系统时钟)			
例	时隙 1	时隙 2	时隙 3	时隙 4	时隙 5	时隙 6	时隙 7	时隙 8	2	4	8	16
1	TB1								TB1			
2	TB1	TB2	TB1	TB2	TB1	TB2	TB1	TB2		TB1-2		
3	TB1	TB2	TB1	TB3	TB1	TB2	TB1	TB3		TB1	TB2-3	
4	TB1	TB2	TB3	TB4	TB1	TB2	TB3	TB4			TB1-4	
5	TB1	TB2	TB3	TB4	TB1	TB2	TB3	TB5			TB1-3	TB4-5
6	TB1	TB2	TB3	TB4	TB1	TB2	TB5	TB6			TB1-2	TB3-6
7	TB1	TB2	TB3	TB4	TB1	TB5	TB6	TB7			TB1	TB2-7
8	TB1	TB2	TB3	TB4	TB5	TB6	TB7	TB8				TB1-8

图 15

计时器系统集成电路 280

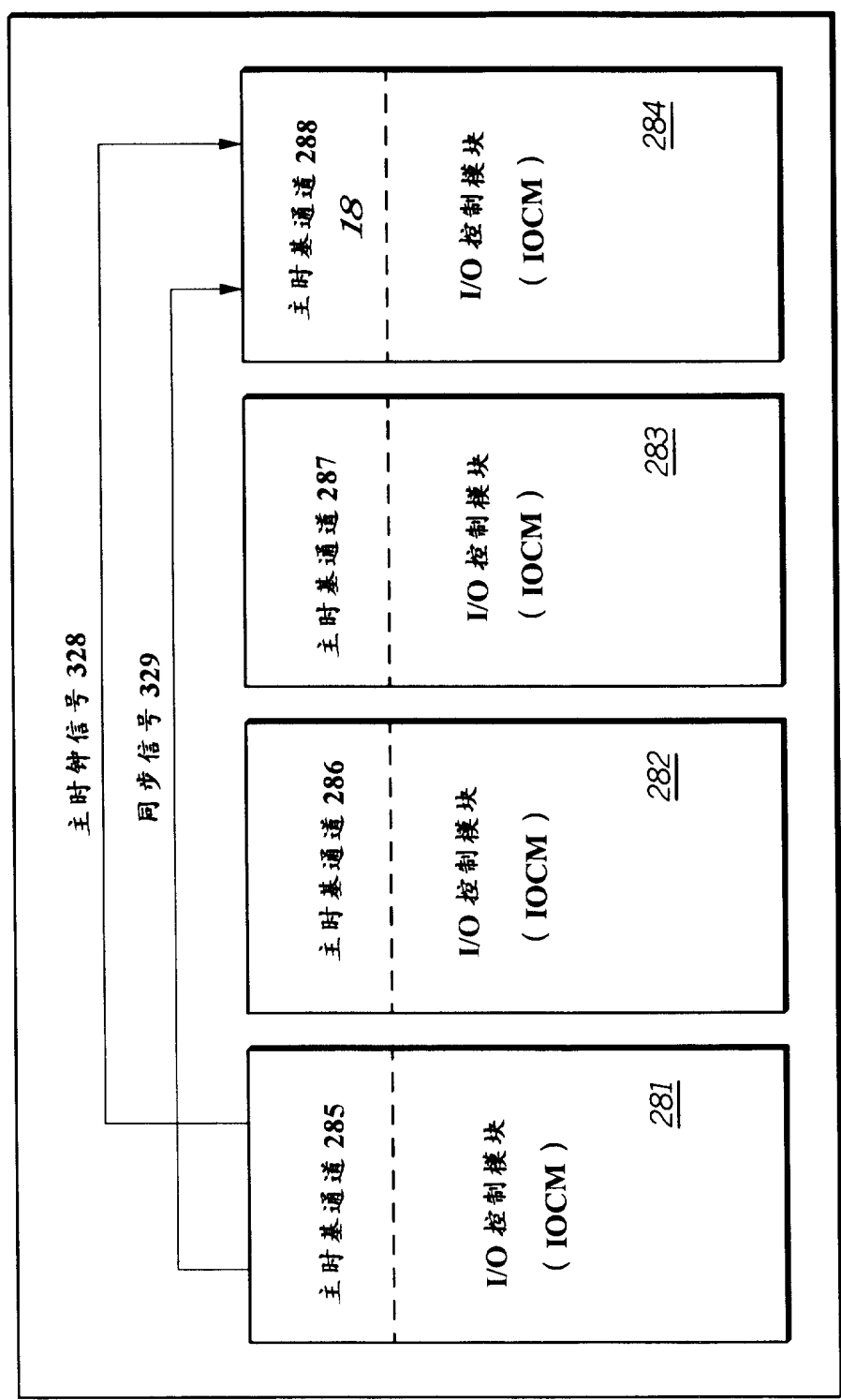


图 16

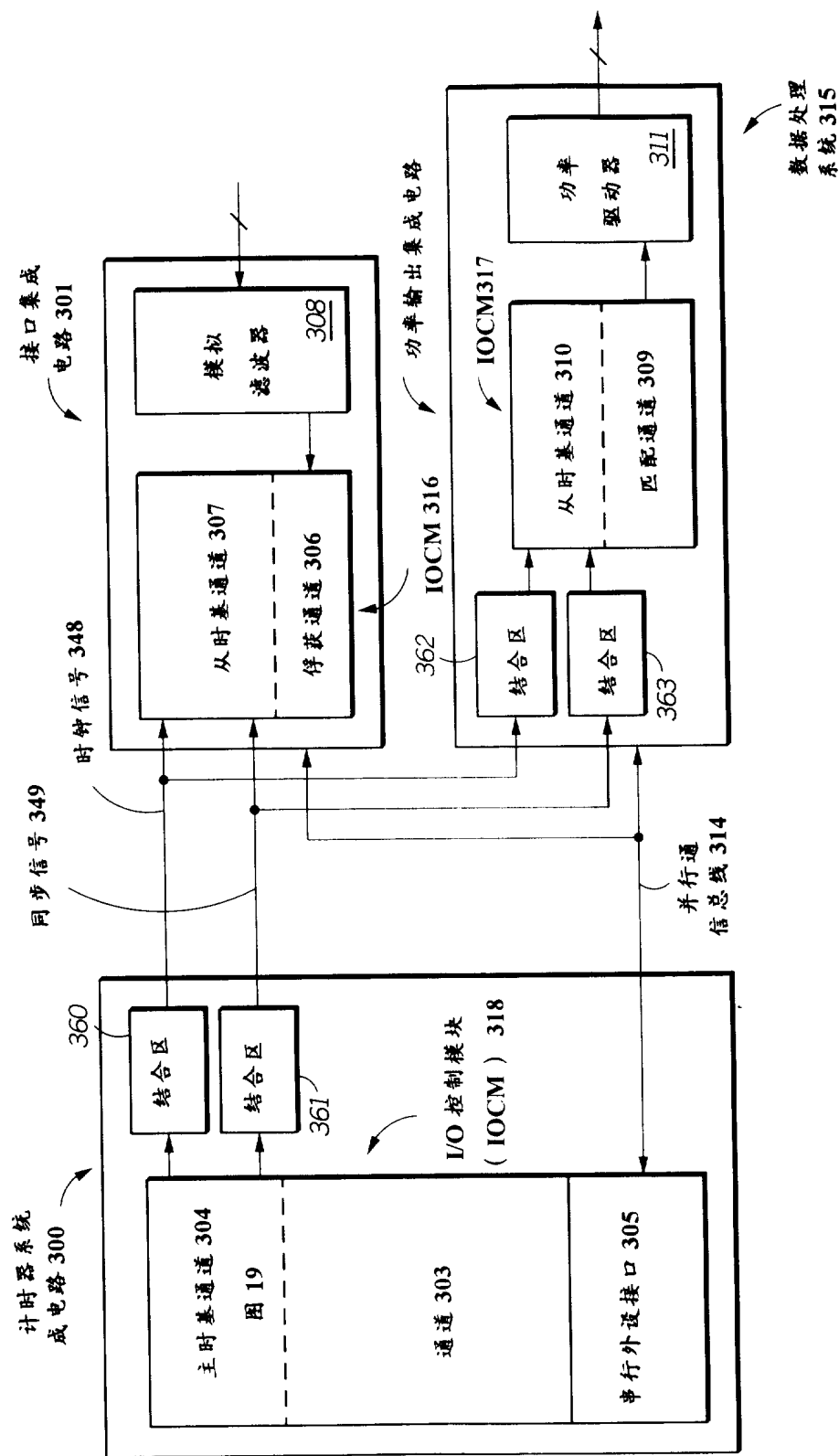


图 17

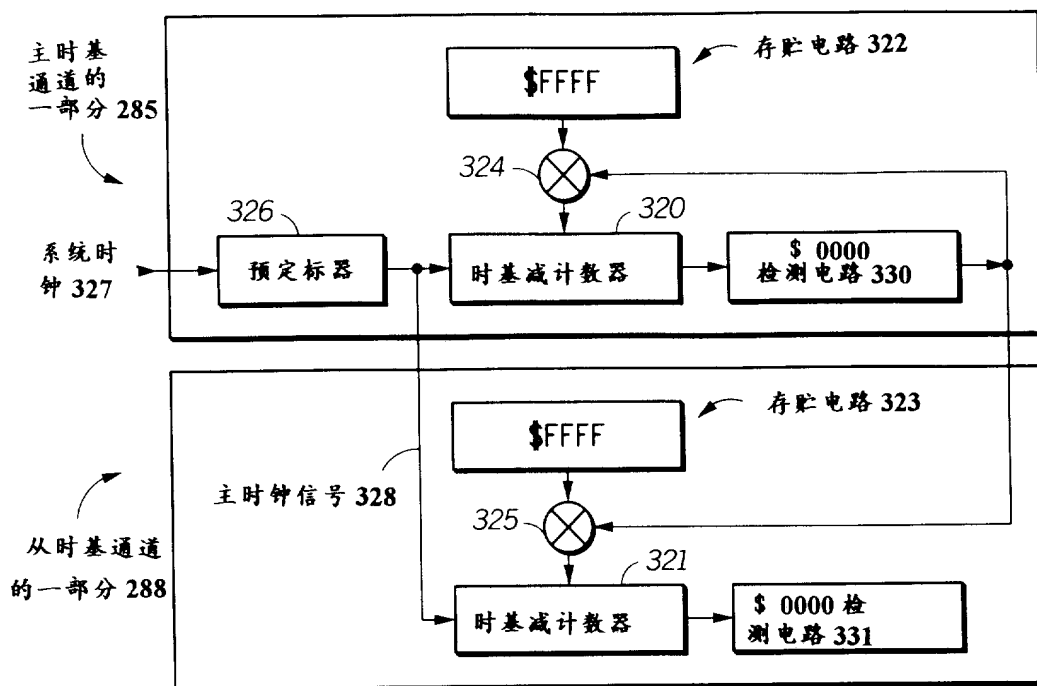


图 18

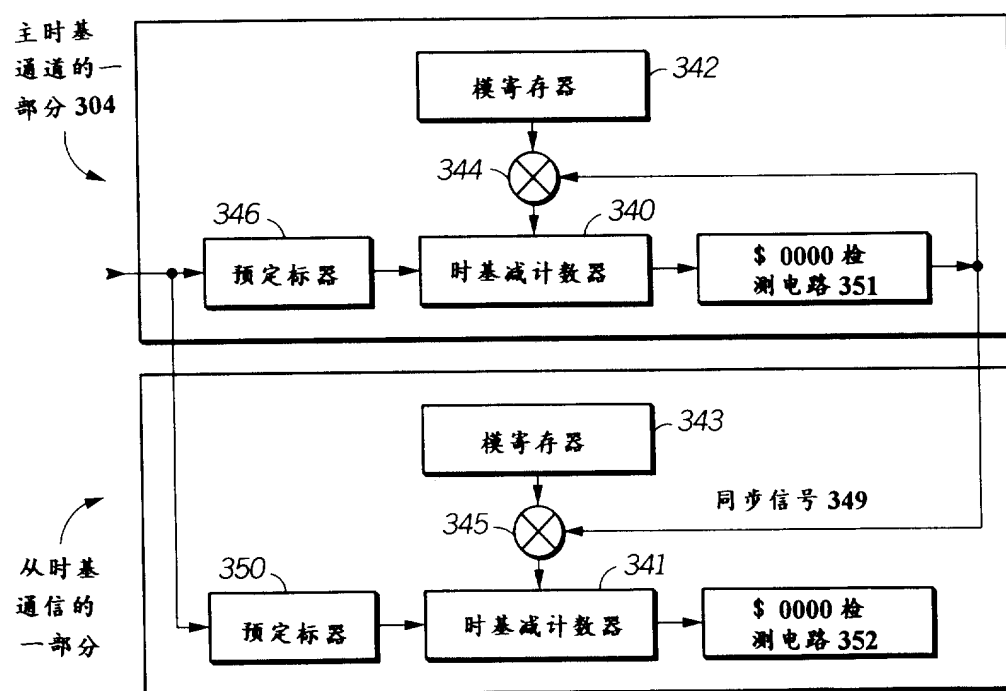


图 19

用 1 中 25 电路一部分

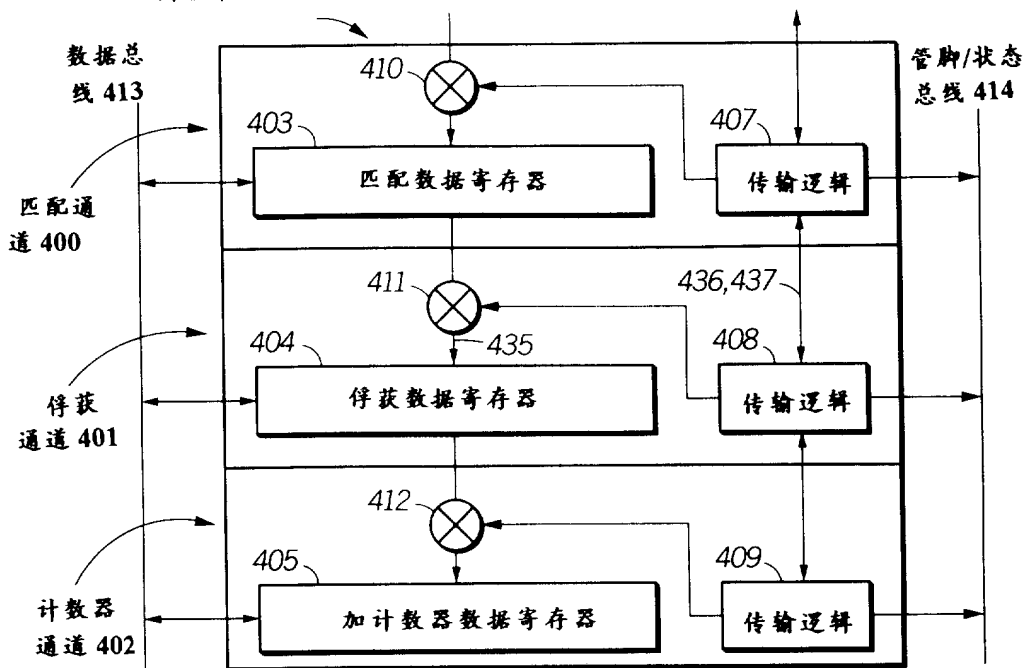
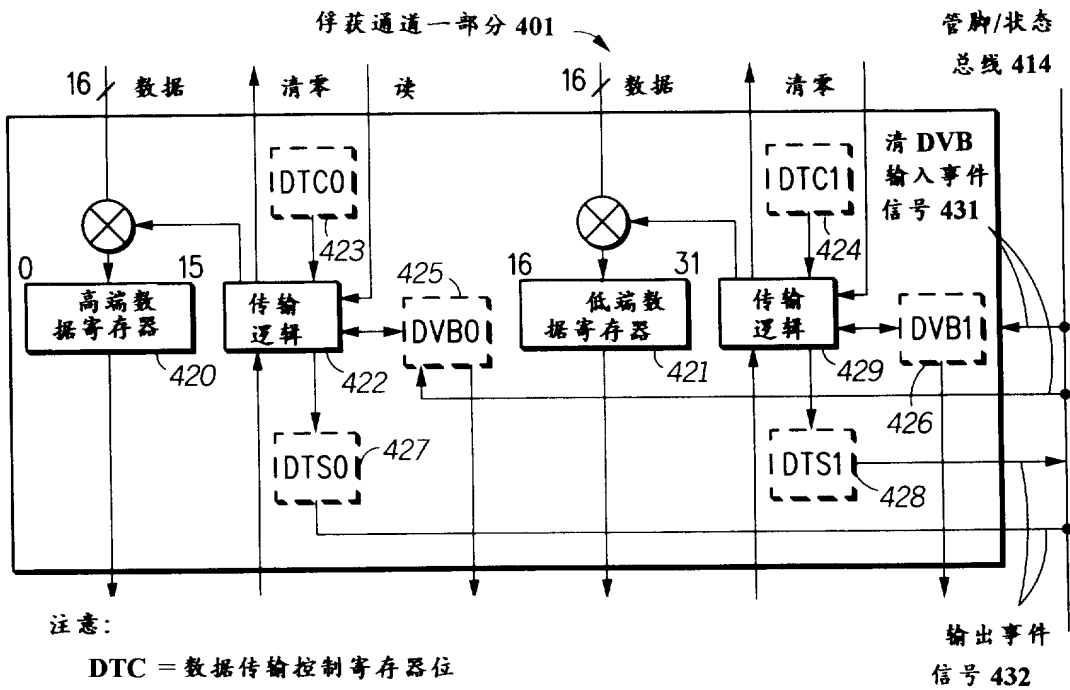


图 20

俘获通道一部分 401



注意：

DTC = 数据传输控制寄存器位

DVB = 数据有效寄存器位

DTS = 数据传输状态寄存器位

VADP/LADP = 有效相邻数据对/无效相邻数据对输出事件信号

图 21

一部分俘获通道 401

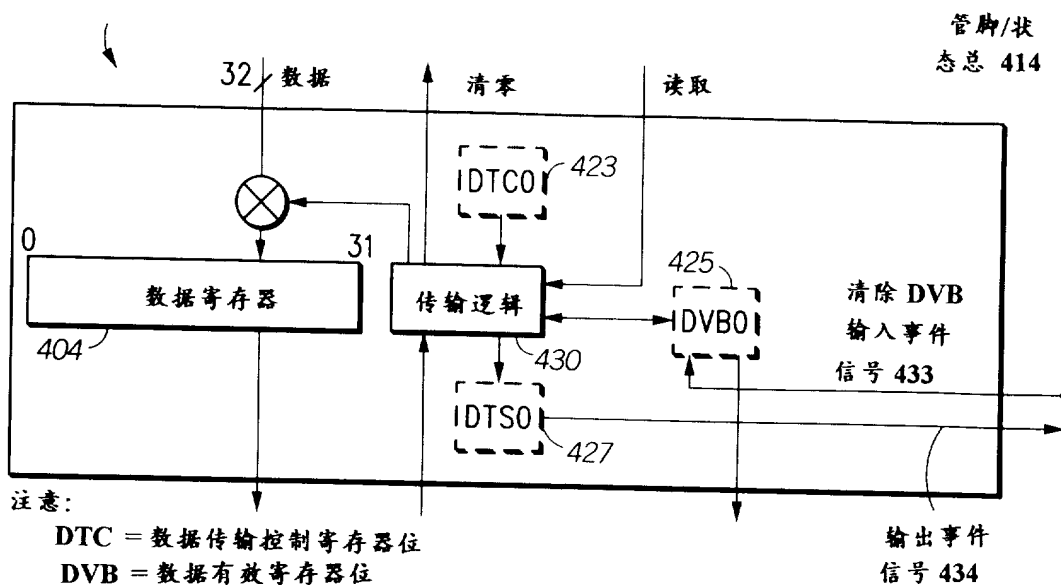


图 22

通信类型	模式	数据有效比特设置 (DVB = 1)			数据有效比特清除 (DVB = 0)		
		写入数 据寄存器	从顶端 相邻通 道传输	操作 结束	读 1 写 0 至 DVB	从底端 相邻通 道传输	操作结束
匹配	匹配	X			X	X	X
	传输	X	X		X	X	
	匹配 传输	X	X		X	X	X
俘获	俘获	X		X	X	X	
	传输	X	X		X	X	
加计数器	计数	X		X	X	X	
	传输	X	X		X	X	
减/加 (双向) 计数器	计数/传输			加计数器	X	X	

图 23

数据传 输控制比特的 逻辑电平 (DTC)	对应数据传输操作
0 0	无效
0 1	没有输出事件有效
1 0	使能, 有效相邻数据对检测输出事件
1 1	使能, 无效相邻数据对检测输出事件

图 24

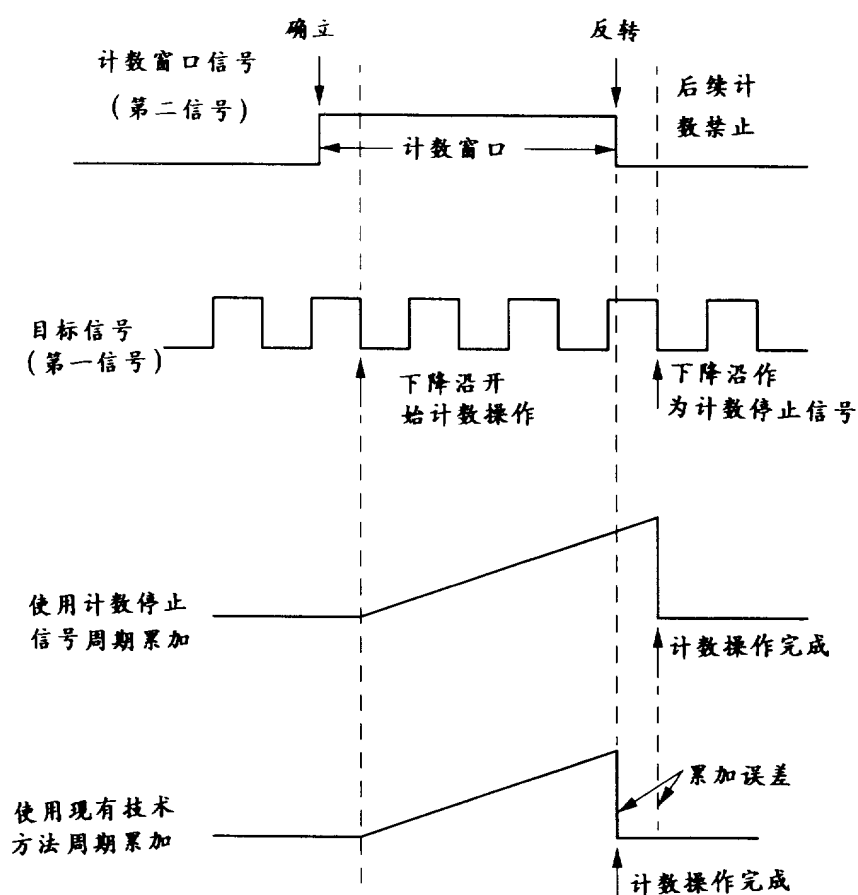


图 25

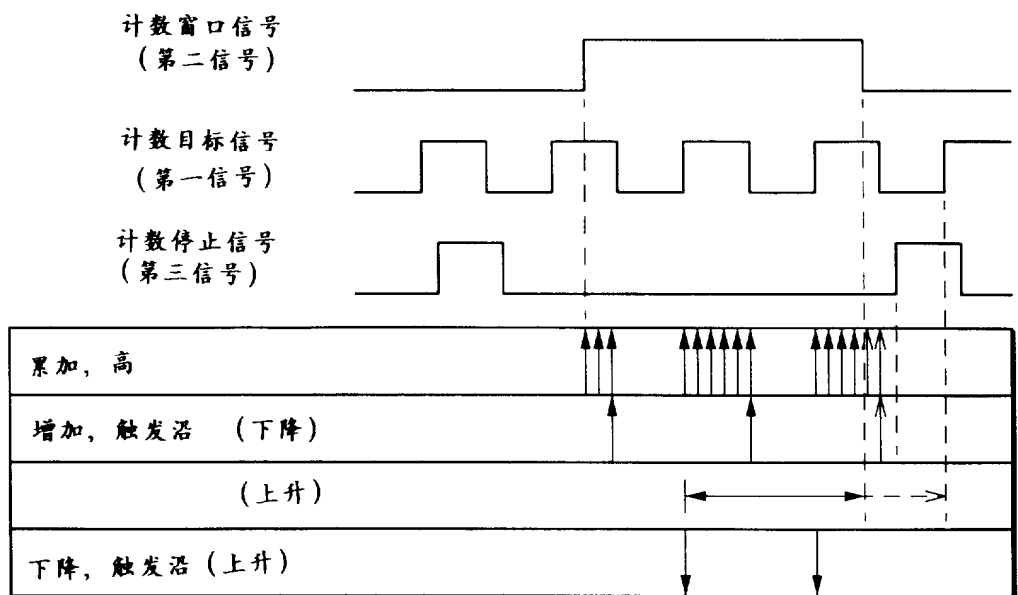


图 26

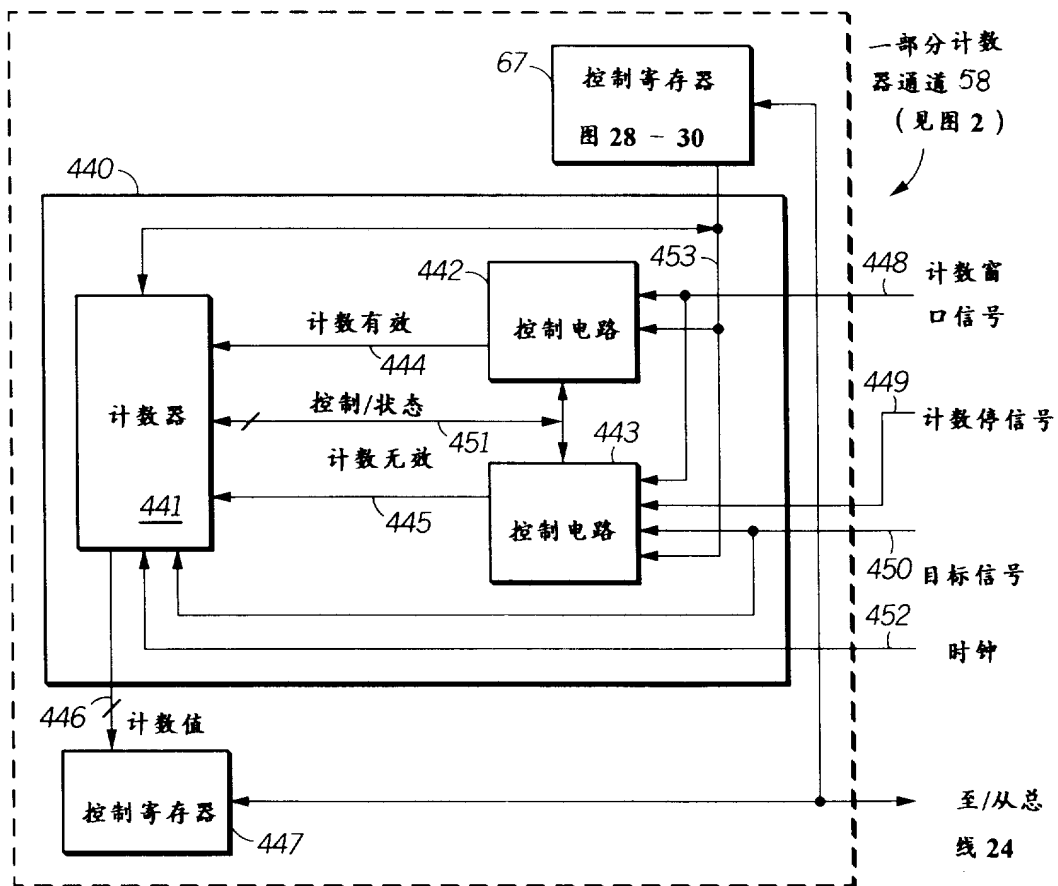
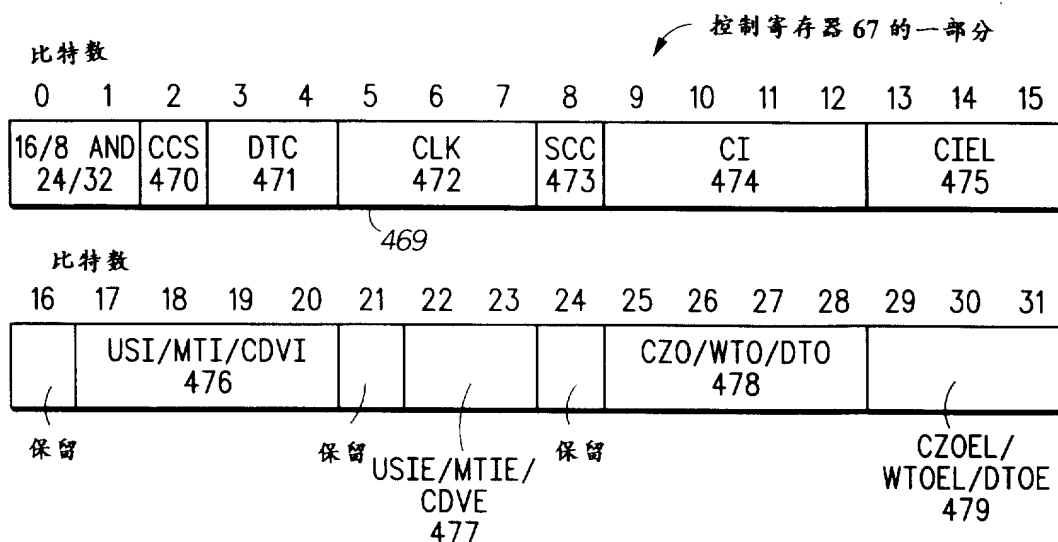


图 27



16/8 和 24/32 - 16 - 比特, 8 - 和 24 - 比特, 或 32 - 比特操作

CCS: 计数器时钟选择

DTC: 数据传输控制

CLK: 时钟源选择

SCC: 单个/连续计数操作

CI: 计数输入

CIEL: 计数输入边沿或电平

USI/MTI/CDVI: 加计数停止输入/
模传输输入/

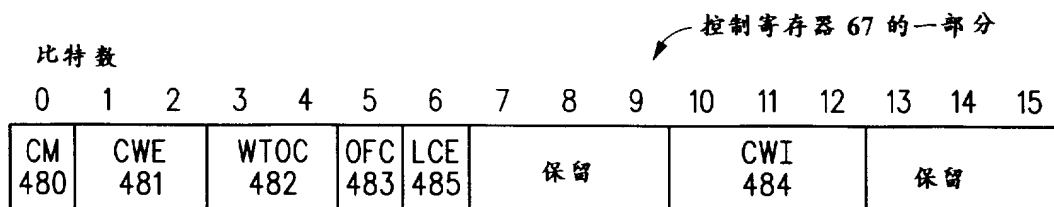
清除数据有效输入

USIE/MTIE/CDVE: 加计数停止输入边沿/
模传输输入边沿/
清除数据有效边沿

CZO/WTO/DTO: 计数到零输出/
窗口终端输出/
数据传输输出

CZOEL/WTOEL/DTOE: 计数到
零输出边沿或逻辑操作/
窗口终端输出边沿或逻辑操作/
数据传输输出边沿

图 28



CM: 计数器模式

CWE: 计数窗口使能

WTOC: 窗口终端输出控制

OFC: 溢出控制

CWI: 计数窗口输入

图 29

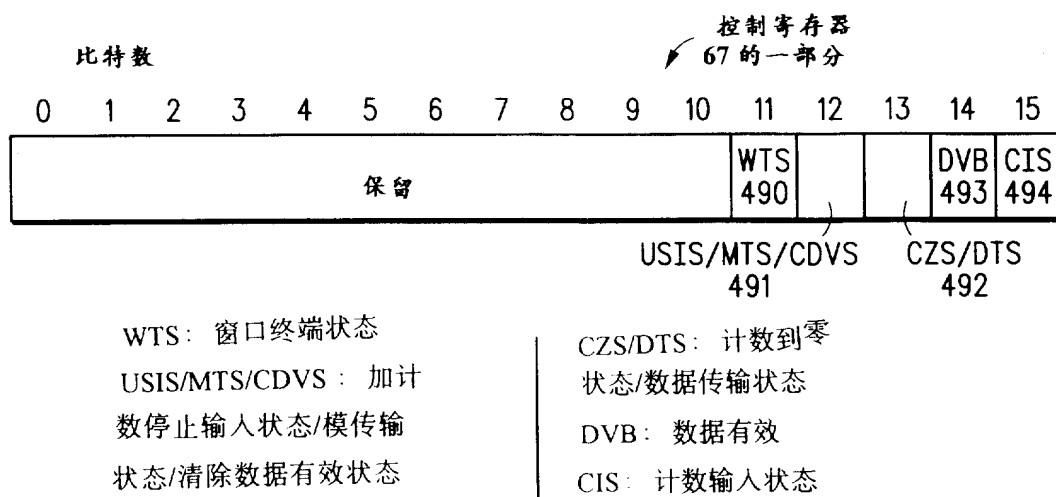


图 30

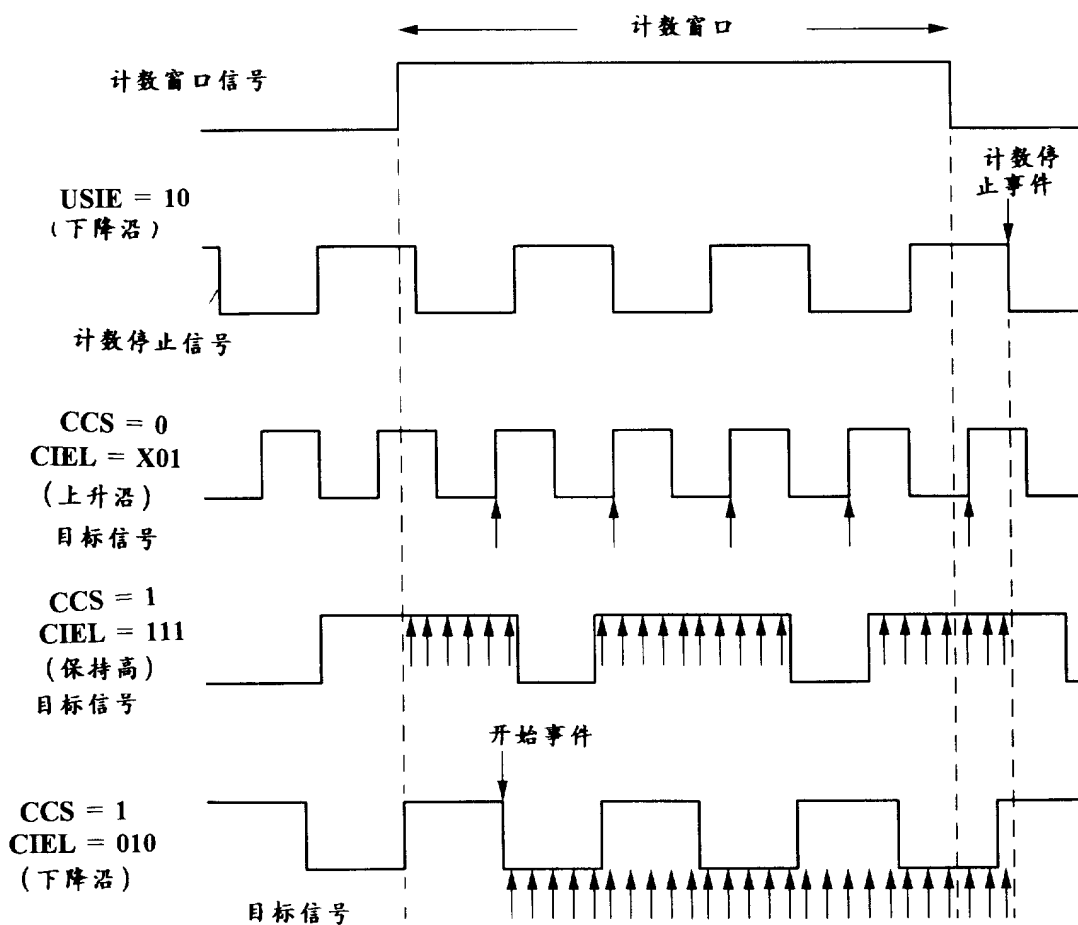


图 31

总线 74



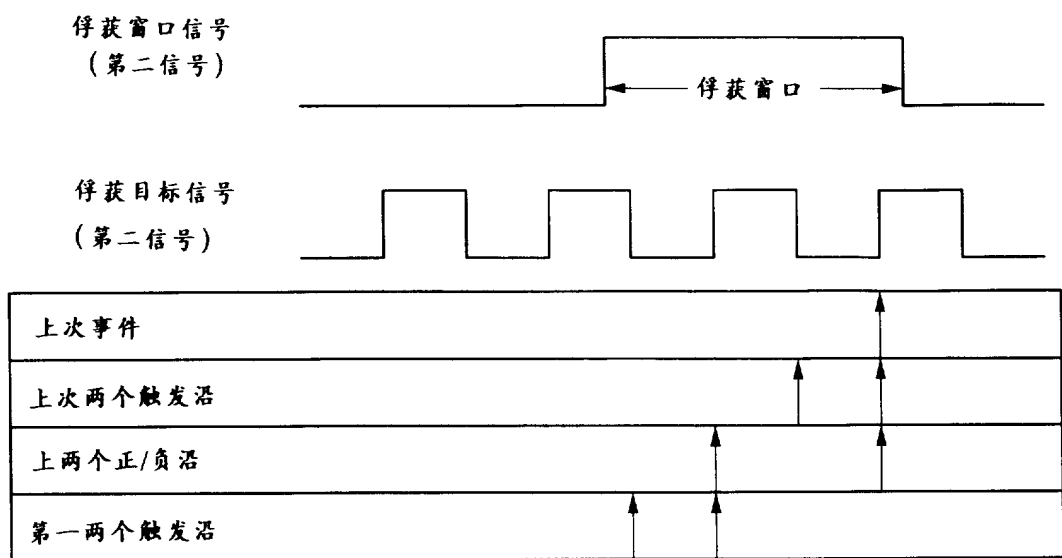


图 33

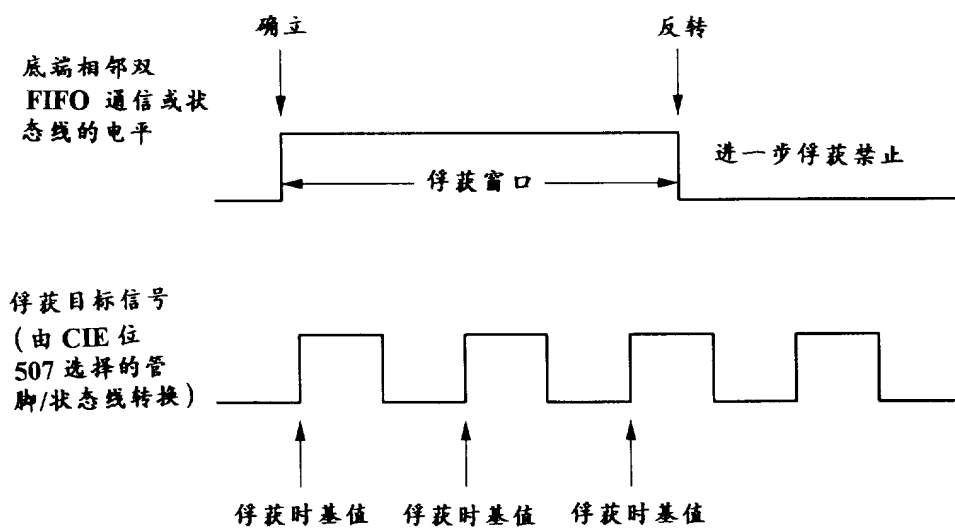


图 34

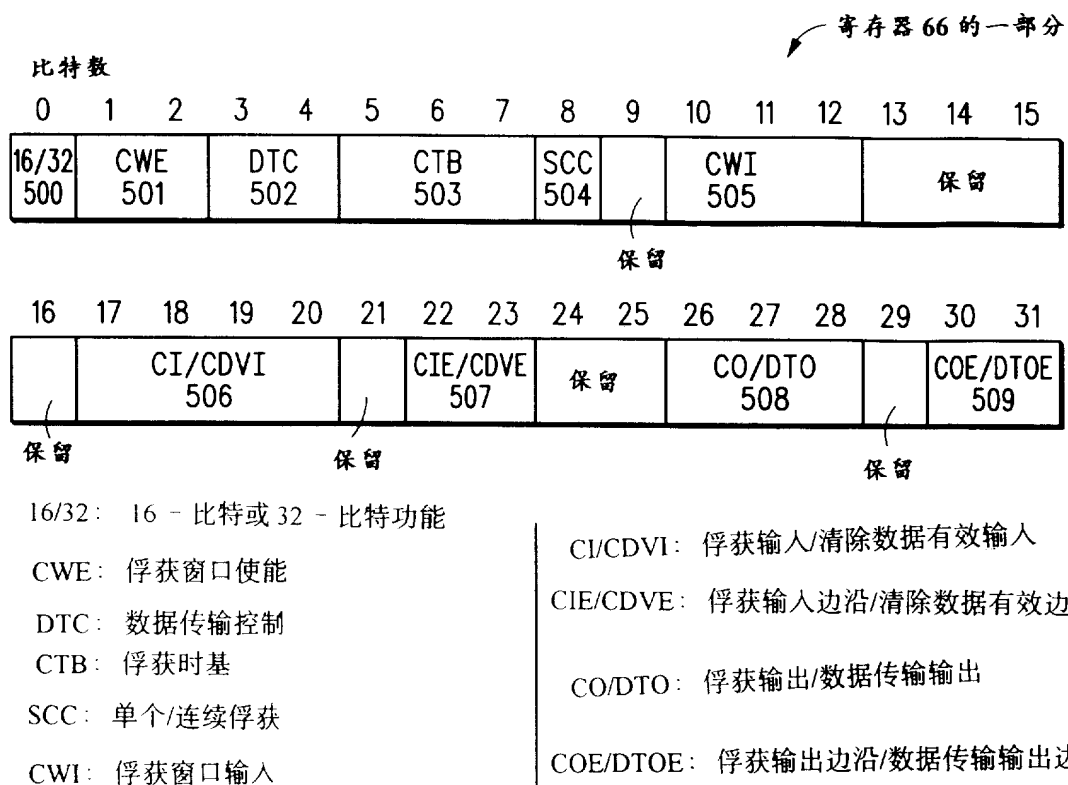


图 35

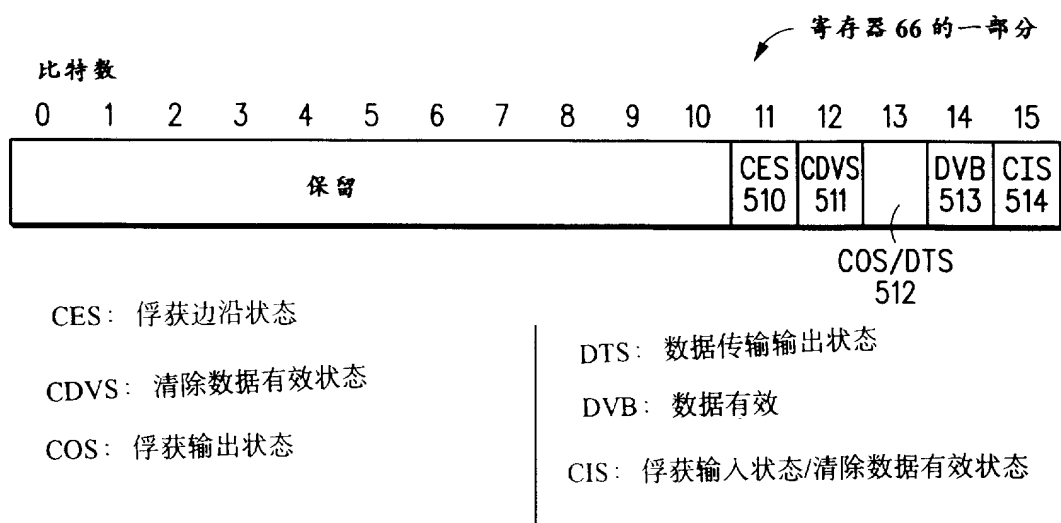


图 36

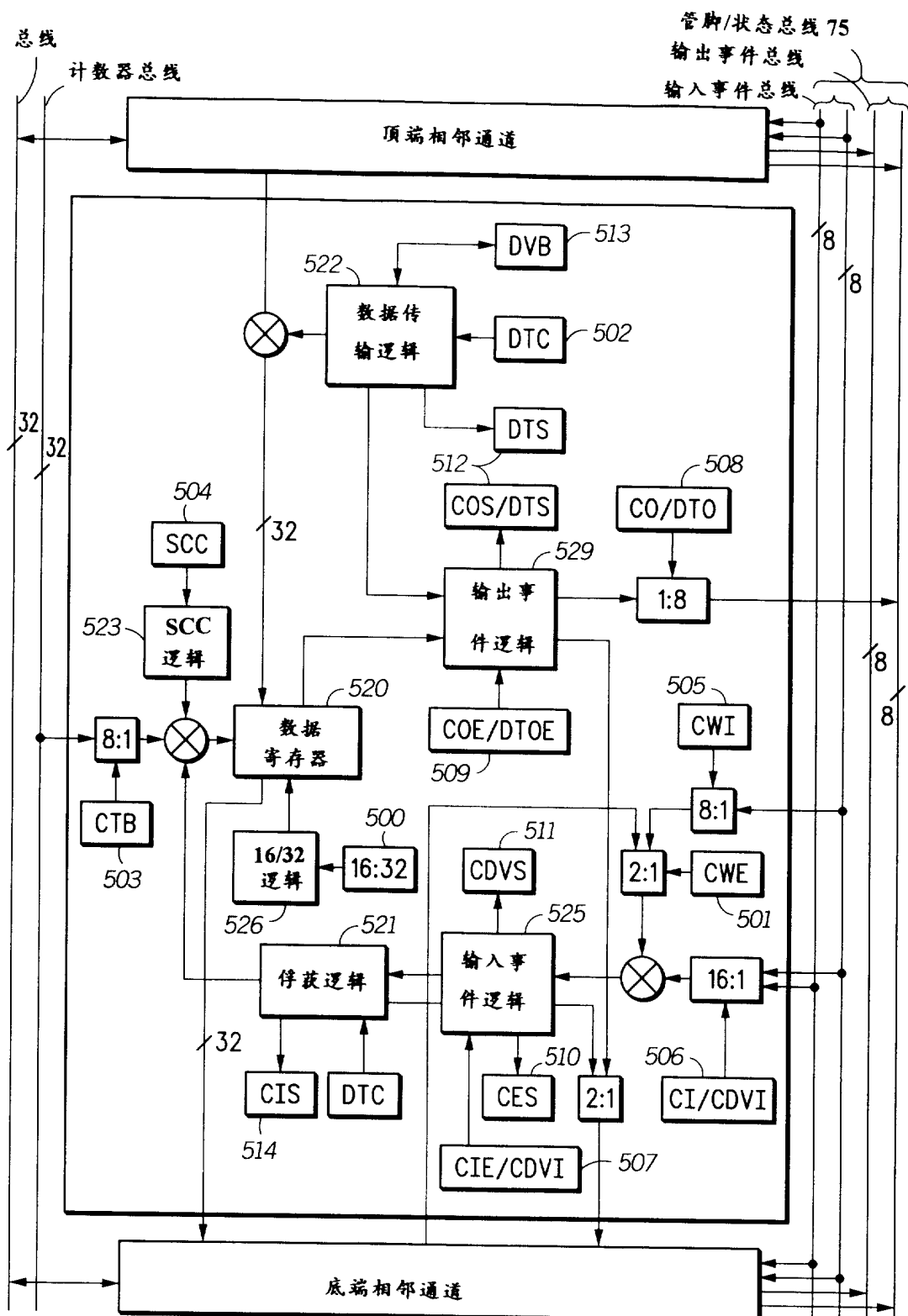


图 37

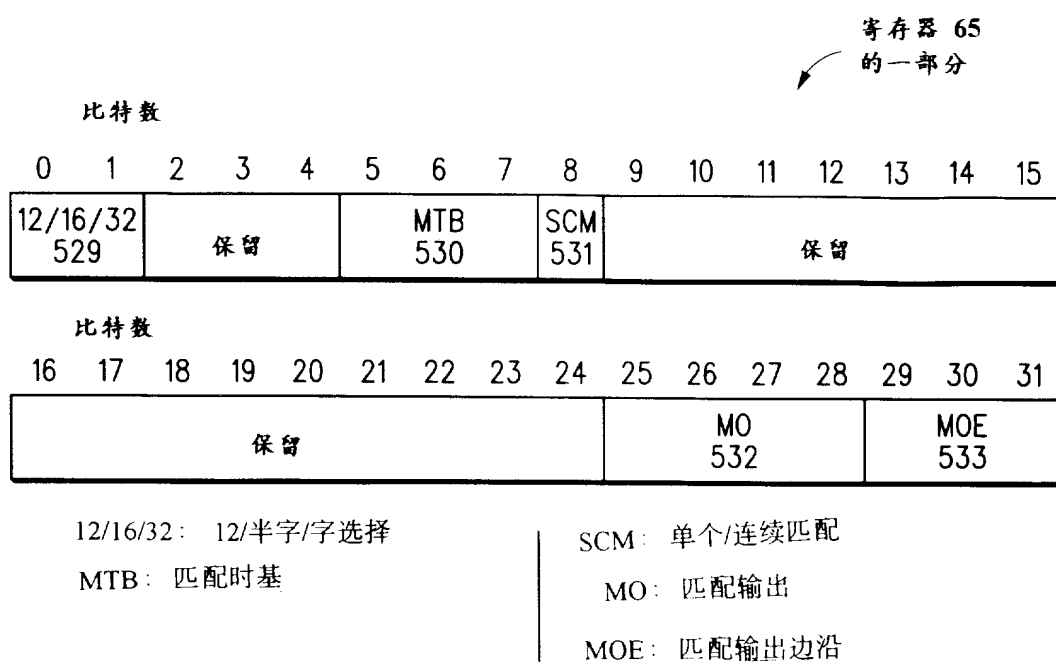


图 38

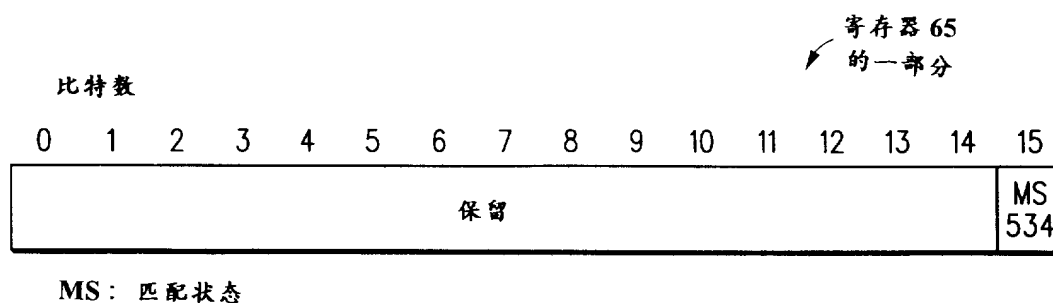


图 39

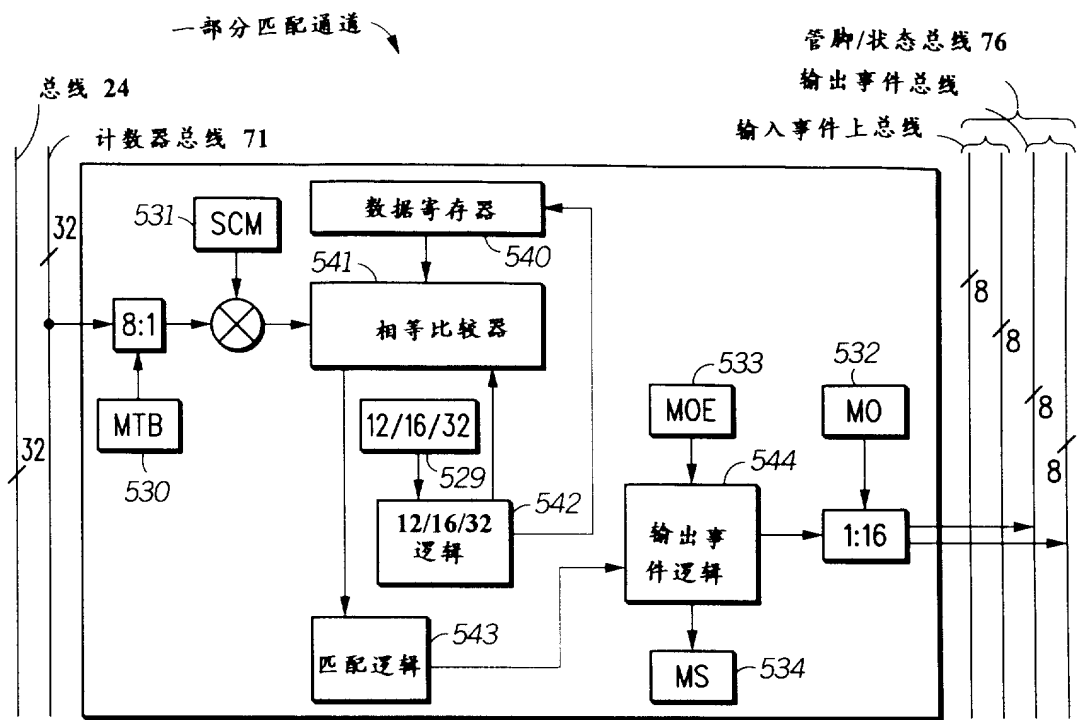


图 40