[19]中华人民共和国国家知识产权局

[51]Int. Cl[7]
G06F 15/00 G11C 19/00

[12]发明专利说明书

[21] ZL 专利号 95119287.6

[45]授权公告日 2002 年 11 月 27 日

[11]授权公告号 CN 1095130C

[22]申请日 1995.11.15 [21]申请号 95119287.6

[30]优先权

[32]1994.12.16 [33]US [31]357909

[73]专利权人 摩托罗拉公司

地址 美国伊利诺斯

[72]发明人 利奥尼德·斯莫伦斯凯 谢·科瓦尔
阿温那·戈伦 大卫·加兰蒂

[56]参考文献

JP58161193 1983. 9.24 G11C11/14

US4839791 1989. 6.13 G06F13/00

US5115810 1992.10.13 G06F1/00

审查员 林松岭

[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

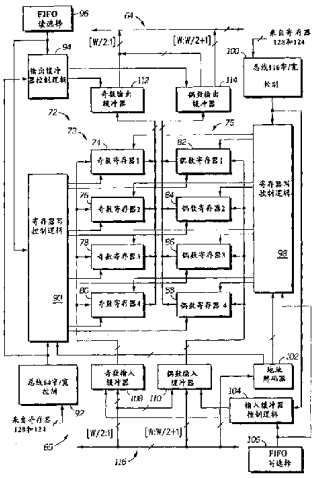
代理人 陆立英

权利要求书 1 页 说明书 20 页 附图 5 页

[54]发明名称 用于可变宽度数据转移的可调深度/宽度先进先出缓冲器

[57]摘要

可调深度/宽度 FIFO 缓冲器(65)适合可变宽度数据的转移,具有两个独立地受控的写寄存器(73、75)部分,用于转移 16 或 32 位字,转移 16 位字时不浪费缓冲器(65)的寄存器空间。当缓冲器(65)置穿转移 16 位字时,存储空间加深。当接口 16、32 位并行数据进,这使缓冲寄存器(72)利用率最大。缓冲器(65)仅从属于主处理器,不能起始数据输出,保持设计简单的和小型。



ISSN1008-4274

1.一种具有先进先出缓冲器的数据处理系统（20），其特征在于，该先进先出缓冲器（72），包括:

多个存储单元（72），该多个存储单元被组织成为第一部分（73）和第二部分（75），该多个存储单元（72）的每个存储单元都具有耦合到第一总线（116）的多个输入端，和耦合到第二总线（64）的多个输出端；和

一个控制逻辑电路（98），耦合到多个存储单元（72），用于控制对第一部分（73）和对第二部分（75）的访问;

其中所述的多个存储单元（72）响应于控制逻辑电路（98），具有第一宽度和第一深度，独立地访问所述的第一和第二部分（73、75），和所述的多个存储单元（72）还响应于控制逻辑电路（98），具有第二宽度和第二深度，共同访问第一和第二部分（73、75），以及其中所述的第一深度或第二深度被提供，这取决于所提供的第一宽度或第二宽度。

2.根据权利要求1的数据处理系统（20），其特征在于，所述的第一深度是所述的第二深度的两倍，和所述的第一宽度等于所述第二宽度的一半。

3.根据权利要求1的数据处理系统（20），其特征在于，所述的多个存储单元（72）作为多个读/写寄存器来表征。

4.根据权利要求1的数据处理系统（20），其特征在于，所述的第一宽度等于32比特和所述第一深度等于4个字，以及所述的第二宽度等于16比特和所述的第二深度等于8个字。

用于可变宽度数据转移的可调深度/宽度先进先出缓冲器

发明技术领域:

本发明涉及先进先出(*FIFO*)存储器,具体涉及用于可变宽度数据转移的可调深度/宽度 *FIFO* 缓冲器。

技术背景:

数字信号处理是对有规律间隔取样的和数字化的读时间信号的算术处理。数字信号处理器(*DSP*)执行诸如信号滤波、混合和比较的数字信号处理功能。在某些数据处理系统中,*DSP* 可包括一个主处理器,用以应付数字信号处理杂务。主处理器可包括例如一个微计算机或微处理器。

偶然地,可能需要相当大量的数据在主处理器与 *DSP* 之间转移。当数据在两个处理器之间转移时因两个处理器之间不兼容,例如两个处理器可能与不同的时钟一起工作,故直接转移不可能。在这种情况下,可以使用异步操作的缓冲存储器。在该数据被接收处理器接收之前先由发送处理器以脉冲串形式传送到缓冲存储器。然而,缓冲存储器可能需要集成电路上大的有效表面区域,对于实施来说是昂贵的,并因所有的数据在该数据可被处理之前必须被转移,故减慢了处理的速度。

为了避免使用相当大的缓冲存储器,有时使用一种较小的先进先出(*FIFO*)缓冲器,以使主处理器与接收处理器相接口。*FIFO* 缓

冲器是一种存储器单元，该单元按照与被接收的数据相同的次序输出数据。数据可能从主处理器的总线传送到 *FIFO* 缓冲器，并且同时 *FIFO* 缓冲器把该数据 传送到接收处理器的总线。*FIFO* 缓冲器用于代替缓冲存储器。该数据可以由许多作为数据流而被传送的数据字组成。每个数据字具有一定数量的比特。例如，一个数据字可以是 16 位比特宽，或是 32 比特位宽。支持的数据传送的 *FIFO* 缓冲器通常支持最宽数据转移。然而，如果利用 32 比特位宽的 *FIFO* 缓冲器仅转移 16 位比特数据，则只有该 *FIFO* 缓冲器的存储容量的一半被利用。

发明内容:

据此，这里按照本发明的一种形式提供一种具有先进先出缓冲器的数据处理系统，该先进先出缓冲器具有多个寄存器和控制逻辑。该多个寄存器的每个寄存器具有耦合到第一总线的多个输入端，和耦合到第二总线的多个输出端。该控制逻辑耦合到多个寄存器。当来自第一总线的数据字具有第一预定宽度时，该控制逻辑允许多个寄存器中的一个寄存器从第一总线接收数据。当来自第一总线的数据字具有第二预定宽度时，该控制逻辑还允许多个寄存器的两个寄存器。

附图说明:

本发明的这些特点和其他特点及其优点通过结合以下附图阅读下文的详细描述将会一目了然。

图 1 示出根据本发明的数据处理系统的方框图。

图 2 示出图 1 的数据处理系统的主接口的一个实施例的方框图。

图 3 示出 根据本发明的一个实施例的图 2 的主接口的可调深度/宽度 *FIFO* 缓冲器的方框图。

图 4 示出图 2 的主接口的 *DSP* 控制寄存器的方框图。

图 5 示出图 2 的主接口的主接口控制寄存器的方框图。

图 6 示出图 2 的主接口的 *DSP* 状态寄存器的方框图。

图 7 示出图 2 的主接口的地址组态配置寄存器的方框图。

图 8 示出根据本发明的在数据流中插入地址的流程图。

具体实施例:

总的来说,本发明提供一种用于可变宽度数据转移的可调深度/宽度 *FIFO* 缓冲器。该 *FIFO* 缓冲器具有两个部分,该两部分的每一部分都独立地受控,以允许 16 位比特字或 32 位比特字要被转移,当 16 比特字被转移时,不浪费 *FIFO* 缓冲器中的存储空间。当取样具有 32 位比特宽度字的外部总线时,该 *FIFO* 缓冲器被组织成为 32 比特位宽和 4 个字深,当取样 16 位比特宽的数据字时,它被组织成为 16 位比特宽和 8 个字深。为此,当接口具有 16 位比特宽的外部总线时,存储区被加深了。在其他实施例中,*FIFO* 缓冲器能够适合用于具有不同宽度的字和能够具有不同的深度。当接口 16 位并行数据或 32 位并行数据时,这允许最大限度利用 *FIFO* 缓冲器的存储空 间。*FIFO* 缓冲器是只从属于主处理器的一种从属缓冲器。*FIFO* 缓冲器不能起始对保持设计简单且小的数据进行取样。

当涉及分别描述信号、状态比特、或者类似的设备进入其逻辑“真”或逻辑“假”状态时,将要使用术语“断定”和“否定”。如果逻辑“真”状态是数字逻辑电平“ 1”,则逻辑“假”状态将是数字逻辑电平“0”。而如果逻辑“真”状态是数字逻辑电平“0”,则逻辑“假”状态将是数字逻辑电平“1”。术语“总线”将用于涉及多种信号,这些信号可能用于转移一种或多种类型的信息,诸如数据、地址、控制或状态。

下面参照图1—8更全面地描述本发明。图1—7中的每一个方框表示一个电路，图8中的每一个方框表示流程图中的一个或多个步骤。图1示出本发明的数据处理系统20的方框图。在图1所说明的实施例中，数据处理系统20是一个数字信号处理器(*DSP*)并设置在一个单片集成电路上。在其它实施例中，数据处理系统20可能是例如一个微计算机或微处理器。数据处理系统20包括定时器22、主接口24、增强型串行同步接口(*ESSI*)26、串行异步接口(*SCI*)28、程序*RAM*(随机存取存储器)和指令超高速缓冲存储器30、数据存储器32、数据存储器34、地址产生单元直接存储器存取(*DMA*)控制器36、外部地址总线开关38、内部数据总线开关40、*DRAM*(动态随机存取存储器)和*SRAM*(静态随机存取存储器)总线接口和指令超高速缓冲存储器控制42、外部数据总线开关44、程序控制单元(*PCU*)46、和数据算术逻辑单元(*ALU*)54。程序控制单元46包括程序中断控制器48、程序解码控制器50、和程序地址发生器52。

标号为"*YAB*"的地址总线56、标号为"*XAB*"的地址总线57、标号为"*PAB*"的程序地址总线58、和标号为"*DAB*"的地址总线59被连接在地址发生单元/*DMA*控制器36与外部地址总线开关38之间。标号为"*DDB*"的数据总线60被耦合在主接口24与外部数据总线开关44之间。标号为"*YDB*"的数据总线61、标号为"*XDB*"的数据总线62、标号为"*PDB*"的程序数据总线63，和标号为"*GDB*"的程序数据总线64被耦合在内部数据总线开关40与外部数据总线开关44之间。

定时器22包括三个定时器，它们可以用于内部或外部的定时，

和可以中断数据处理系统 20 或外部设备的信号。此外，定时器 22 在已经发生规定的事件数目以表示 *DMA* 转移。三个定时器的每一个被耦合到单一的双向插头或端子。定时器 22 的每个定时器还被耦合到总线 57、总线 59、程序中断控制器 48 和到总线 60。

主接口 24 为数据处理系统 20 与其他设备（诸如微计算机、微处理器、或 *DMA*）之间的通信提供双向接口。主接口 24 经总线 60 还双向耦合到外部数据总线开关 44，双向耦合到全局数据总线 64，到程序中断控制器 48，经由总线 57 和 59 到地址产生单元/*DMA* 控制器 36，到外部地址总线开关 38。主接口 24 还从主处理器双向耦合到 50 个外部插头或端子，用于双向数据转移、地址寄存器选择和控制通信。

增强型串行同步接口（*ESSI*）26 耦合到 12 个双向外部插头，以提供与包括例如一个或多个工业标准编译码器、*DSP*（数字信号处理器）、或微处理器的各外部串行设备的串行通信。*ESSI*26 还具有端子用以耦合到总线 57、总线 59、和总线 60。

串行通信接口（*SCI*）28 耦合到 3 个双向外部插头，用于提供与外部设备的串行通信。*SCI*28 还具有耦合到总线 57、总线 59 和总线 60 的端子。

图 1 的数据处理系统 20 的实施例具有三个存储器空间：程序 *RAM* 和指令超高速缓冲存储器 30、*X* 存储器 32 和 *Y* 存储器 34。程序 *RAM* 和指令超高速缓冲存储器 30 耦合到地址总线 58 和数据总线 63。*X* 存储器 32 耦合到地址总线 57、地址总线 59、数据总线 60 和列数据总线 62。*Y* 存储器 34 耦合到地址总线 56、地址总线 59、数据总线 60、和到数据总线 61。

地址发生单元/*DMA* 控制器 36 耦合到地址总线 56、57、58 和 59。地址发生单元/*DMA* 控制器 36 向定时器 22、主接口 24、*ESSI*26、*SCI* 28、程序 *RAM* 和指令高速缓冲存储器 30、存储器 32、存储器 34、外部地址总线开关 38、和 *DRAM* 与 *SRAM* 总线接口和指令超高速缓冲存储器控制 42 提供存储器地址。在一个优选实施例中，该 *DMA* 控制器具有 6 个信道。

DRAM 与 *SRAM* 总线接口和指令超高速缓冲存储器 42 耦合到程序地址总线 58 和 14 个双向外部插头。*DRAM* 的指令超高速缓冲存储器和 *SRAM* 总线接口和指令高速缓冲存储器 42 起到在外部主存储器(未示出)与程序控制单元 46 之间的缓冲存储器的作用。指令超高速缓冲存储器存储经常要利用的程序指令。当由程序要求的指令字可以从高速缓冲存储器中得到时，由于消除了访问主存储器所需要求的时间，所以可使性能得到提高。

内部数据总线开关 40 耦合到数据总线 60、数据总线 61、数据总线 62、程序数据总线 63，和全局数据总线 64。外部数据总线开关 44 经由数据总线 60、数据总线 61、数据总线 62、程序数据总线 63，和全局数据总线 64 耦合到内部数据总线开关 40。此外，外部数据总线开关 44 经由数据总线 60 耦合到定时器 22、主接口 24、*ESI*26、和 *SCI*28。内部数据总线开关 40 用于在各总线之间进行转移。任何两个总线可被通过内部总线开关 40 彼此连接。外部地址总线开关 38 和外部数据总线开关 44 分别耦合各外部总线(未示出)到任何内部地址和内部数据总线。

在程序控制单元 46 中，程序中断控制器 48 对各个中断请求进行判断，和被耦合到定时器 22、主接口 24、*ESSI*26、和 *SCI*28。另外，

程序中断控制器48被双向耦合到全局数据总线64和程序解码控制器50。程序解码控制器50解码每个24比特的指令和被双向耦合到程序中断控制器48和到程序地址发生器52。程序地址发生器52包括所有用于程序地址发生、系统堆栈、和环路控制需要的硬件。此外，程序地址发生器52耦合到程序地址总线58和到程序数据总线63。

数据算术逻辑单元(*ALU*)54耦合到程序数据总线63、数据总线61、和数据总线62。数据*ALU*54对数据操作数执行全部算术和逻辑操作。*ALU*54包含有通过总线61和62的方式可以读出或写入的寄存器。数据*ALU*54还耦合到总线63和到总线60。

时钟发生电路(未示出)向图1所示的所有方框提供时钟信号。在数据处理系统20中还有测试电路，在图1中没示出。

图2示出图1的数据处理系统20的主接口24的方框图。主接口24包括主发送数据*FIFO*缓冲器65、*DSP*发送数据*FIFO*缓冲器66、主接口控制寄存器67、*DSP*状态寄存器68、*DSP*控制寄存器69、和地址配置寄存器70。在所说明的实施例中，主接口24具有两个*FIFO*缓冲器。主发送数据*FIFO*缓冲器65具有多个耦合到外部总线116的数据输入端，多个耦合到全局数据总线(*GDB*)64的数据输出端、耦合到主接口控制寄存器67的多个输出端的第一多个控制端，和耦合到*DSP*控制寄存器69的多个输出端的第二多个控制端。*DSP*发送数据*FIFO*缓冲器66具有多个耦合到全局数据总线64的数据输入总线，多个耦合到外部总线116的数据输出总线，耦合到主接口控制寄存器67的多个输出端的第一多个控制端，和耦合到*DSP*控制寄存器69的多个输出端的第二多个控制端。

主接口控制寄存器67还具有耦合到外部总线116的多个双向端子，和耦合到*DSP*状态寄存器68的多个输出端子。*DSP*状态寄存器68具有耦合到全局数据总线64的多个输出端，和*DSP*控制寄存器69具有耦合到全局数据总线64的多个双向端子，和耦合到*FIFO*缓冲器65与66的控制端的多个输出端。

主处理器(未示出)通过主接口24到全局数据总线64的方式可以初始化从总线116到数据处理系统20中的一个位置的数据转移。外部总线116和总线64两者是双向总线。在一个实施例中，总线116是传送数据与地址两者的复用总线。在另外的实施例中，外部总线116可以具有单独的数据线和地址线。当要求从总线116向总线64转移时，主处理器控制该转移。*FIFO*缓冲器65仅作为从属于主处理器的从属来操作。可由主处理器写入*FIFO*缓冲器65，但是不能由*FIFO*缓冲器65起始转移。同样，*FIFO*缓冲器66是从属，只从数据处理系统20向外部总线116的发送数据。

主接口控制寄存器67(图5)和*DSP*控制寄存器69(图4)为确定总线116和总线64之间的数据转移提供控制比特。主处理器还通过主接口控制寄存器67与*DSP*状态寄存器68通信，启动*FIFO*缓冲器65，确定*FIFO*缓冲器65与66是空还是满，和用于接收如在下面讨论图6时所描述的其他状态信息的类型。地址组态配置寄存器70耦合到总线116和包括用于编程基本地址的地址比特，以确定访问*FIFO*缓冲器65的地址范围。

图3示出根据本发明的一个实施例的主接口24的可调深度/宽度*FIFO*缓冲器65的方框图。*FIFO*缓冲器65包括*FIFO*寄存器72、寄存器读控制逻辑90、总线64窄/宽控制92、输出缓冲器控

制逻辑 94、*FIFO* 读选择 96、寄存器写控制逻辑 98、总线·116 宽/宽控制 100、地址解码器 102、输入缓冲器控制逻辑 104、*FIFO* 写选择 106、奇数输入缓冲器 108、偶数输入缓冲器 110、奇数输出缓冲器 112 和偶数输出缓冲器 114。总线 116 是用以连接输出缓冲器 112 和 114 的双向总线。*FIFO* 寄存器 72 包括寄存器部分 73 和寄存器部分 75。寄存器部分 73 包括奇数寄存器 74、76、78 和 80。寄存器部分 75 包括偶数寄存器 82、84、86 和 88。

在一个优选实施例中，*FIFO* 缓冲器 65 包括图 1 的主接口 24 的一部分。另外，*FIFO* 缓冲器 66 可以具有类似于 *FIFO* 缓冲器 65 的电路。数据被从总线 116 提供到 *FIFO* 缓冲器 65。总线 116 是一个外部总线，它可能是主处理器（未示出）的一部分。总线 64 是至主接口 24（图 2）的一个内部总线。在其他的实施例中，总线 116 与 64 可以是任何两个之间需要缓冲的总线。可能需要缓冲，例如是因为两个总线受到不同时钟的控制。

总线 116 和 64 转移具有 *W* 位的数据字，其中 *W* 是一个整数。奇数输入缓冲器 108 具有耦合到总线 116 用于接收数据比特〔*W*/2:1〕的多个输入端，耦合到每个奇数寄存器 74、76、78 和 80 的各个输入端的多个输出端，和一个用于接收来自输入缓冲器控制逻辑 104 的控制信号的控制端。偶数输入缓冲器 110 具有耦合到总线 116 用于接收数据比特〔*W*:*W*/2+1〕的多个输入端，耦合到每个偶数寄存器 82、84、86 和 88 的多个输入端的多个输出端，和一个用于接收来自输入缓冲器控制逻辑 104 的控制信号的控制端。奇数输出缓冲器 112 具有耦合到奇数寄存器 74、76、78 和 80 的多个输出端的多个输入端，耦合到总线 64 用于提供数据比特〔*W*/2:1〕的多

个输出端，和一个耦合到输出缓冲器控制逻辑 94 的一个输出端用于接收控制信号的控制端。偶数输出缓冲器 114 具有耦合到每一个偶数寄存器 82、84、86 和 88 的多个输出端的多个输入端，耦合到总线 64 用于提供数据比特〔*W*:*W*/2+1〕的多个输出端，和一个耦合到输出缓冲器控制逻辑 94 的输出端的控制端。

FIFO 读选 择 96 具有耦合到输出缓冲器控制逻辑 94 的输入端的输出端。*FIFO* 读选择 96 的输出端还耦合到寄存器读控制逻辑 90 的输入端。寄存器读控制逻辑 90 具有耦合到奇数寄存器 74、76、78 和 80 以及偶数寄存器 82、84、86 和 88 的每一个的输入端的输出端，用于提供读控制信号。在一个优选实施例中，奇数寄存器 74、76、78 和 80 以及偶数寄存器 82、84、86 和 88 的每一个是常规的 16 位读/写寄存器。寄存器读控制逻辑 90 和寄存器写控制逻辑监视 *FIFO* 寄存器 72，以跟踪 *FIFO* 寄存器 72 是空的还是满的。

总线 6 4 窄/宽控制 92 具有耦合到图 2 的寄存器 128 和 124 的多个输入端，和耦合到输出缓冲器控制逻辑 94 和寄存器读控制逻辑 90 两者的输入端的输出端。寄存器写控制逻辑 98 具有耦合到奇数寄存器 74、76、78 和 80 以及偶数寄存器 82、84、86 和 88 的每一个的输入端的输出端，用于提供写控制信号。地址解码器 102 具有耦合到总线 116 用于接收多个地址信号的多个输入端，和耦合到寄存器写控制逻辑 98 的多个输出端。

FIFO 写选择 106 具有耦合到输入缓冲器控制逻辑 104 的输入端和到寄存器写控制逻辑 98 的输入端和输出端。当主处理器写数据到 *FIFO* 缓冲器 65 时，*FIFO* 写选择 106 把一个启动信号提供到输入缓冲器控制逻辑 104 和到寄存器写控制逻辑 98。总线 116

窄/宽控制 100 具有耦合到寄存器 124 和 128 的多个输入端，和耦合到寄存器写控制逻辑 98 的输入端和到输入缓冲器控制逻辑 104 的输入端的输出端。输入缓冲器控制逻辑 104 具有连接到奇数输入缓冲器 108 的输入端的第一输出端，和连接到偶数输入缓冲器 110 的输入端的第二输出端。

数据被从总线 116 写入 *FIFO* 缓冲器 65，和数据被从 *FIFO* 缓冲器 65 读到总线 64。*FIFO* 寄存器 72 被分为两部分，部分 73 和 75。部分 75 是以偶数寄存器 82、84、86 和 88 为特征的“偶数”部分，和部分 73 是以奇数寄存器 74、76、78 和 80 为特征的“奇数”部分。在另外的实施例中，可以有任何数目的部分，这取决于有多少种不同的数据宽度要被容纳。每个部分可以独立地写入和读出。总线 116 和 64 两者具有动态的变尺寸，其中各种尺寸的字可以在同一总线上传送。在该说明的实施例中，*FIFO* 缓冲器 65 可以接收具有高达 32 位宽度的字宽(*W*)的数据，和 *FIFO* 缓冲器 65 具有深度(*D*)，它是可变的，这取决于字的宽度(*W*)。

对于已说明的 *FIFO* 缓冲器 65 的实施例，对于转移数据存在着四种可能性：1)*FIFO* 缓冲器 65 可以从总线 116 接收 16 位字和向总线 64 提供 16 位字；2)*FIFO* 缓冲器 65 可以从总线 116 接收 16 位字和向总线 64 提供 32 位字；3)*FIFO* 缓冲器 65 可以从总线 116 接收 32 位字和向总线 64 提供 16 位字；和 4)*FIFO* 缓冲器 65 可以从总线 116 接收 32 位字和向总线 64 提供 32 位字。如果 16 位字被转移，*FIFO* 寄存器 72 是 8 个字深。如果 32 位字被转移，*FIFO* 寄存器 72 是 4 个字深。对于下面数据转移的例子，通过 *FIFO* 缓冲器 65，*W* 等于 32。但是，在其他实施例中，*W* 可能等于

其他数据宽度。

当从总线116向*FIFO*缓冲器65写入16位字和读16位字到总线64时，两个总线116和64视为一种窄的16位宽*FIFO*缓冲器。总线116窄/宽控制100限定总线116为窄，即*W*/2比特宽。总线64窄/宽控制92限定总线64为窄，即*W*/2比特宽。*FIFO*为选择106提供控制信号到寄存器写控制逻辑98和到输入缓冲器控制逻辑104，使*FIFO*缓冲器65做好从总线116取样数据的准备。来自总线116的输入数据被按照比特〔*W*/2:1〕取样到奇数输入缓冲器108或到偶数输入缓冲器110。来自奇数输入缓冲器108的第一个16比特字根据〔*W*/2:1〕写入按照由地址解码器102从总线116接收的地址确定的奇数寄存器74、76、78和80之一。地址解码器102提供解码地址到寄存器读控制逻辑90。寄存器读控制逻辑90选择奇数寄存器74、76、78或80之一有待写入。第二16位数据利用偶数缓冲器110根据比特〔*W*/2:1〕写入偶数寄存器82、84、86和88之一。第三16位数据字根据比特〔*W*/2:1〕写入奇数寄存器74、76、78和80之一。这种转移通过偶数和奇数寄存器之间的交替持续进行直至完成，或者*FIFO*满。

当从*FIFO*缓冲器65读出一个16位数据时，16位数据字被从奇数寄存器74、76、78和80或从偶数寄存器82、84、86、和88之一输出。第一16位字被从奇数寄存器74、76、78和80之一读出，通过奇数输出缓冲器112，输出比特〔*W*/2:1〕。第二16位字被从偶数寄存器82、84、86和88之一读出，通过偶数输出缓冲器114，输出比特〔*W*/2:1〕。第三16位字被从奇数寄存器74、76、78和80之一读出，通过输出缓冲器112，输出比特〔*W*/2:1〕。这种操作持续下

去，直至转移完成或 *FIFO* 缓冲器 65 空。在该所说明的实施例中，每个奇数寄存器或偶数寄存器是 16 位宽。在其他的实施例中，各寄存器可以是任何宽度。

当从总线 116 写一个 16 位数据到 *FIFO* 缓冲器 62 和读 32 位字到总线 64 时，总线 116 察觉 *FIFO* 缓冲器 65 作为具有 8 个字的深度和总线 64 察觉 *FIFO*65 作为具有 4 个字的深度。总线 64 窄/宽控制 92 确定总线 64 作为宽的，即 *W* 位宽。*FIFO* 写选择 106 提供控制信号到寄存器写控制逻辑 98 和到输入缓冲控制逻辑 104，为 *FIFO* 缓冲器 65 从总线 116 取样数据做准备。从总线 116 的输入数据根据比特〔*W*/2:1〕取样到奇数输入缓冲器 108 或偶数输入缓冲器 110。数据不是根据〔*W*:*W*/2+1〕比特取样的。在一个优选择实施例中，*W* 等于 32。从奇数输入缓冲器 108，第一 16 位数据字根据〔*W*/2:1〕写入按照由地址解码器 102 从总线 116 接收的地址确定的奇数寄存器 74、76、78 和 80 之一。地址解码器 102 提供一个解码的地址到寄存器读控制逻辑 90。寄存器读控制逻辑 90 选择奇数寄存器 74、76、78、和 80 之一有待写入。第二 16 位数据字利用偶数输入缓冲器 110 根据〔*W*/2:1〕写入偶数寄存器 82、84、86、和 88 之一。第三 16 位数据字根据〔*W*/2:1〕写入奇数寄 器 74、76、78 和 80 之一。该转移通在奇数和偶 数寄存器之间交替持续进行直至完成，或者 *FIFO* 缓冲器 65 已满。

32 位数据字同时从奇数寄存器 74、76、78、和 80 之一以及偶数寄存器 82、84、86 和 88 之一读到总线 64。32 位数据字的 16 位被从寄数寄存器 74、76、78 和 80 读出，通过奇数输出缓冲器 112，输〔*W*/2:1〕位。其他 16 位被从偶数寄存器 82、84、86 和 88 读出，通

过偶数输出缓冲器114，输出〔*W*:*W*/2+1〕位。地址解码器102提供解码的地址到寄存器读控制逻辑90，选择哪个奇数和偶数寄存器从中被读出。这种操作持续进行，直至转移完成或者*FIFO*缓冲器65空。

当从总线116取样32位字到*FIFO*缓冲器65和读16位字到总线64时，总线116察觉*FIFO*缓冲器65作为具有4个字深和总线64察觉FIFO缓冲器作为具有8个字深。总线116窄/宽控制100限定总线116作为宽的，即*W*位宽度，和提供控制信号，使寄存器写控制逻辑98和输入缓冲器控制逻辑102做好准备，接收32位字。总线64窄/宽控制逻辑92确定总线64作为窄的，即*W*/2位宽，和使寄存器读控制逻辑90和输出缓冲控制逻辑94作好写入32位字的准备。*FIFO*写选择106提供启动信号到寄存器写控制逻辑90和到输入缓冲器逻辑104，使*FIFO*缓冲器65从总线116取样数据。地址解码器102提供一个解码地址到寄存器写控制逻辑98。寄存器写控制逻辑98选择奇数寄存器74、76、78或80之一有待写入。32位数据字的16位根据〔*W*/2:1〕从总线116取样到奇数输入缓冲器108，与此同时，其他16位根据〔*W*:*W*/2+1〕位被取样到偶数输入缓冲器110。

当读16位数据字到总线64时，待读的〔*W*/2:1〕的第一个字是从奇数寄存器74、76、78和80之一读到奇数输出缓冲器112的。第二个16位字〔*W*:*W*/2+1〕是从偶数寄存器82、84、86和88之一读到偶数输出缓冲器114的。第三个16位字是从奇数寄存器74、76、78和80之一读的，以此类推，直至转移完成或*FIFO*缓冲器65空。地址解码器102提供解码地址到寄存器读控制逻辑90，选择

哪个奇数和偶数寄存器有待读出。

当从总线 116 取样 32 位字到 *FIFO* 缓冲器 65 和读 32 位字到总线 64 时，总线 116 和 64 察觉 *FIFO* 缓冲器 65 作为具有 4 个字深。总线 116 窄/宽控制 100 确定总线 116 作为宽的，即 *W* 位宽，和提供一个控制信号使寄存器写控制逻辑 98 和输入缓冲器控制逻辑 102 做好准备，接收 32 位字。总线 64 窄/宽控制 92 还确定总线 64 作为宽的，和使寄存器读控制逻辑 90 和输出缓冲器控制逻辑 94 做好准备写 32 位字。*FIFO* 写选择 106 提供一个启动信号到寄存器写控制逻辑 98 和到输入缓冲器控制逻辑 104，使 *FIFO* 缓冲器 65 从总线 116 取样 32 位数据字。地址解码器 102 提供一个解码地址到寄存器写控制逻辑 98。寄存器写控制逻辑 98 选择奇数寄存器 74、76、78 和 80 之一根据〔*W*/2:1〕接收 16 位字，与此同时，其他 16 位被根据〔*W*:*W*/2+1〕位取样到偶数输入缓冲器 110。

32 位数据字同时从奇数寄存器 74、76、78 和 80 之一和偶数寄存器 82、84、86 和 88 之一读到总线 64。32 位数据字的 16 位被从奇数寄存器 74、76、78 和 80 读出，通过奇数输出缓冲器 112 输出〔*W*/2:1〕位。其他 16 位被从偶数寄存器 82、84、86、和 88 之一读出，通过偶数输出缓冲器 114 输出〔*W*:*W*/2+1〕位。地址解码器 102 提供解码地址到寄存器读控制逻辑 90，选择哪个奇数和偶数寄存器响应于来自 *FIFO* 读选择 96 的启动信号被读出。这种操作持续进行，直至转移完成，或者 *FIFO* 缓冲器 65 空。

通过分割 *FIFO* 寄存器为各部分，和独立控制该各部分，可变宽度的数据字可以在不浪费 *FIFO* 缓冲器中的存储面积下被转移。因为缓冲器存储面积被更有效地利用，集成电路的面积被有效的利

用。

图 4 示出图 2 的主接口的 *DSP* 控制寄存器 69 的方框图。*DSP* 控制寄存器 69 包括 32 个控制比特当主接口 24 作为总线控制器正在发送数据时，标号“*FC*1”和“*FC*0”的控制比特 130 和 131 在主发送数据 *FIFO* 缓冲器 65 和在 *DSP* 发送数据 *FIFO* 缓冲器 66 中控制数据转移的格式。当主接口 24 正在从总线 116 读或写时，控制位 130 和 131 确定 *FIFO* 缓冲器的数据宽度，在 *FIFO* 寄存器 72 中的数据对准，和符号扩充。标号为“*BL*5—*BL*0”的控制位 132 控制数据长度。在控制位 132 中的值在数据被转移的每个数据转移周期之后被递减。当控制位 132 的值达到 0，或达到另外的预定值，该转移结束。

图 5 示出图 2 的主接口 24 的主接口控制寄存器 67 的方框图。主接口控制寄存器 67 包括控制位 136—142。当主接口 24 正在由主处理器读时分别标号为“ *HRF*1”和“*HRF*0”的主接收数据转移格式控制位 136 和 137 确定 通 过 *DSP* 发送数据 *FIFO* 缓冲器 66 数据转移的数据转移格式。控制位 136 和 137 确定在 *FIFO* 缓冲器中数据宽度，在 *FIFO* 寄存器中的数据对准，和符号扩充。当主接口 24 正在由主处理器写入时，分别标号为“ *HTF*1”和“*HTF*0”的主发送数据转移格式控制位 138 和 139，确定利用主发送数据 *FIFO* 缓冲器 65 转移的数据转移格式。控制位 138 和 139 确定在 *FIFO* 缓冲器中的数据宽度，在 *FIFO* 寄存器中的数据对准，和符号扩充。标号为“*HF*2”、“*HF*1”和”*HF*0”的控制位 140、141 和 142 是用于总线 116 和总线 64 之间通信的一般目的主标志。控制位 140、141 和 142 可以由主处理器(未示出)认定或否定。

图6示出图2的主接口24的*DSP*状态处理寄存器68的方框图。*DSP*状态寄存器68包括比特145—151。标号为"*HACT*"的比特145被认定主接口24总是有效的。当一次转移已完成和在主接口24中的操作已中止，比特145被否定。标号为*HF*2、*HF*1和*HF*0的比特的146、147、和148表示在主接口控制寄存器67中主标志*HF*2、*HF*1、和*HF*0的比特状态。只有主接口可以改变比特146、147和148。标号为"*SRRQ*"的比特149表示主发送数据*FIFO*缓冲器65含有的用于数据处理系统20的数据。当在主接口数据*FIFO*缓冲器65中的各寄存器由数据处理系统20置空时，比特149被否定。标号"*STRQ*"的比特150当断定时，表示在*DSP*发送数据*FIFO*缓冲器66中的各寄存器未满，和可以由数据处理系统20写入。标号为"*HCP*"的比特151当被断定时，表示主命令中断正在悬置。当中断已经被主接口24运行时，比特151被否定。比特151或者可以由主接口24否定，或者由主处理器(未示出)否定。

图7示出图2的主接口24的地址配置寄存器70的方框图。标号"*PM*31—*PM*36"的地址位153确定访问*FIFO*缓冲器65的地址范围。取决于主接口24的操作模式，地址配置寄存器70可以由主处理器访问，或可以由主接口24直接访问。

图8示出根据本发明的一个实施例的插入一个地址到*FIFO*缓冲器65的数据流所要求的步骤的流程图。步骤155和156为判断步骤，和步骤157至161为插入一个地址到数据流中执行的步骤。利用地址的范围经由总线116可以访问*FIFO*缓冲器65。地址范围是由从总线116取样的地址的最高有效位确定的。例如，如果32比特地址的16个最高有效位被用于识别访问*FIFO*缓冲器65，

FIFO 缓 器 65 可被访问 2^{16} 或 65536 个单元。最低有效位“几乎”无须注意，因为 65536 个单元的 7 个是不同于 *FIFO* 缓冲器 65 的寄存器。*FIFO* 缓冲器 65 的输出可以通过若干机制中的任何机制转移到一个地址范围。例如，较低序的地址位可以被用作指示字，控制通过 *FIFO* 缓冲器 65 存储输入数据的位置。

当通过 *FIFO* 缓冲器转移一个数据流时，用于访问 *FIFO* 缓冲器的地址可能丢失，除非做出某些保存该地址的措施。在 *FIFO* 缓冲器 65，从总线 116 首先 取 样的数据是该地址。该地址也可以包含在首部信息中。在地址中的首部信息可以用于控制数据到数据处理系统 20 的所需单元，或数据处理系统如何利用该地址的描述。在所说明的实施例中，总线 116 是一个用于传送地址与数据信息两者的“ 复用总线”。所描述的地址插入方法要求系统部件读数据，例如地址发生单元/*DMA* 控制器 36 识别该地址，诸如该地址被从 *FIFO* 缓冲器 65 读出时。存在若干方法从在复用的总线中的数据中区分地址。例如，当事务处理开始时，如果 *FIFO* 缓冲器空，地址是从 *FIFO* 缓冲器读出的第一个数据。另外，如果数据串是可知的长度，从 *FIFO* 缓冲器读出的下一个字是一个地址。另外，数据流的长度可能是地址的函数，在这种情况下，在一个数据流之后待读的下一个字是一个地址。

当地址信息的总线 116 取样以后，地址解码器 102 提供一个启动信号到寄存器写控制逻辑 98。然后地址被按照从总线 116 第一个取样的数据写入，因为当提供写启动信号时，该地址在总线 116 上。接收地址以后，从总线 116 取样数据。当主接口 24 为从属时，由主处理器控制数据流的大小。当主接口 24 为主导时，由 *DSP*

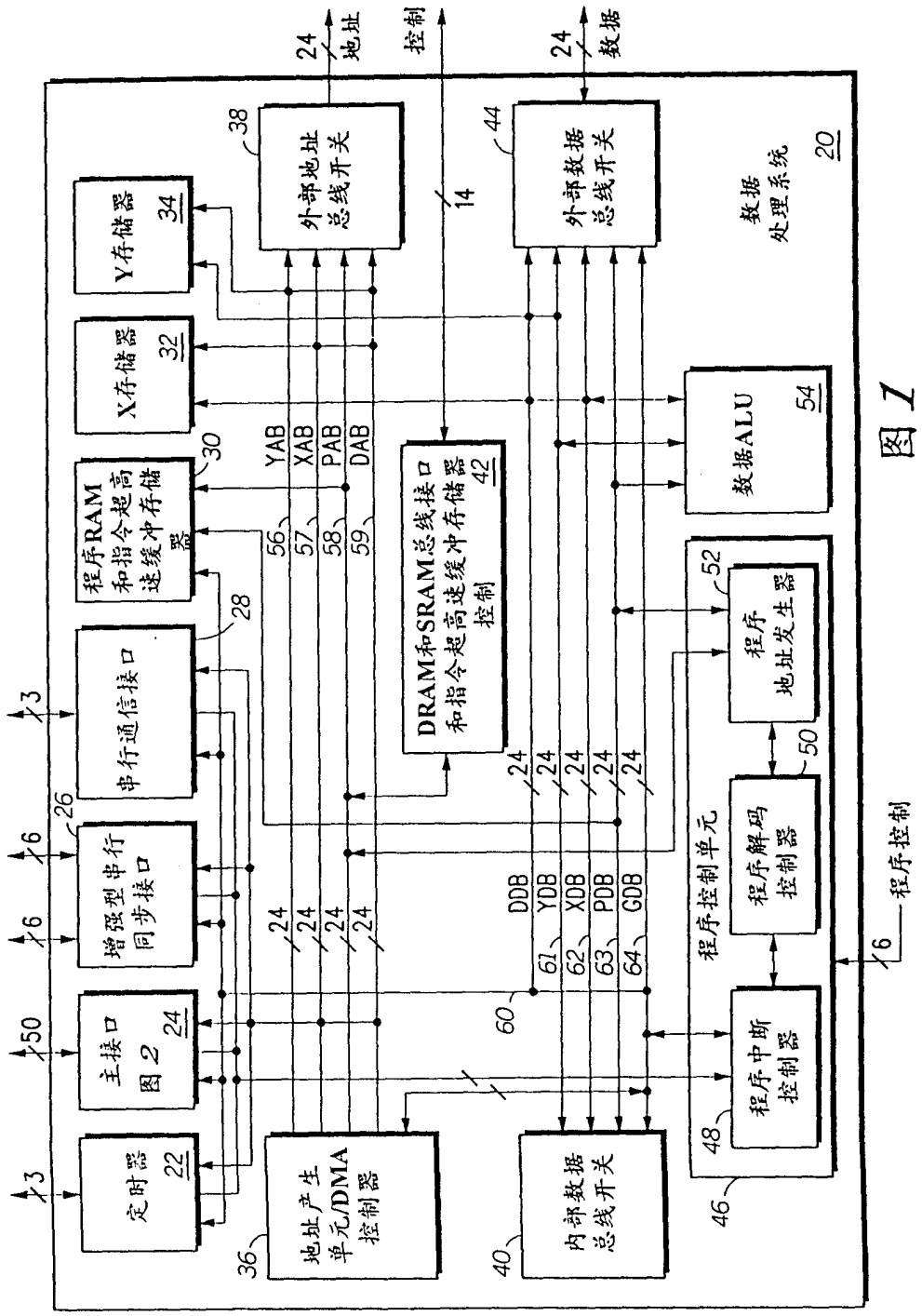
控制寄存器69(图4)的比特132中的初始值确定数据流的大小。每次转移数据字时，逆减比特132的值。当该值等于0时，转移完成。

在判断步骤155中，主处理器确定该地址是否在访问*FIFO*缓冲器65所要求的范围内。如果该地址不在所要求的范围内，采取"否"通路，和在步骤157，取消对*FIFO*缓冲器65的访问。如果该地址在所要求的范围内，则采取判断步骤155的"是"通路。在步骤156中，主处理器通过检验*DSP*状态寄存器68(图6)中的位145，确定是否*FIFO*可以利用。如果*FIFO*缓冲器是不可以利用的，则采取"否"的通路到步骤158，访问*FIFO*缓冲器65被取消。在访问批准之前，主处理器必须等待，直至*FIFO*缓冲器65完成当前的事务处理。如果*FIFO*缓冲器是可利用的，则采用"是"通路，到达步骤159。和由主接口24取样地址到达*FIFO*缓冲器65。注意，该事务处理是利用从总线116经由*FIFO*缓冲器65到总线64描述的。然而，在其他的实施例中，该事务处理也可能是利 用类似的步骤从总线64经由*DSP*发送数据*FIFO*缓冲器66到总线116的。在地址被写入*FIFO*缓冲器66的寄存器以后，在步骤160，一个数据串被写入*FIFO*缓冲器65。一旦数据字被写入*FIFO*缓冲器65，即当*FIFO*缓冲器65不空时，在步骤161，各数据字从*FIFO*缓冲器65被读到数据处理系统20的一个存储位置。当数据流在被写入*FIFO*缓冲器65的同时，数据流可能被从*FIFO*缓冲器65中读出。*FIFO*缓冲器总保持部分空。如果在事务处理期间*FIFO*缓冲器65被充满，可以由主接口24插入等待状态，防止主处理器重写*FIFO*缓冲器65。

总线116被表示为一种复用点线，意味着该总线传送地址和数据信息两者。然而，在另外的实施例中，总线116可以分为单独的地址和数据总线，诸如*ISA*总线。还应当注意，在描述在图7的实施例中，*FIFO*缓冲器65被利用在主接口24中，然而，在其他实施例中，应当利用具有固定深度和宽度的常规*FIFO*缓冲器。

通过利用小的*FIFO*缓冲器代替较大的存储器阵，在数据处理系统中节约了空间。当从一种规模总线到具有不同规模的另一种总线转移数据时，利用可变深度和宽度的*FIFO*缓冲器允许更有效地利用*IFFO*缓冲器的存储区。为了防止通过*FIFO*缓冲器的数据串的地址被丢失，地址被插入在数据流的开始。

当本发明在一个优选的实施例的正文中已经被描述的同时，对于本专业的技术人员来说，本发明可能在许多方面被修改和可以设想出许多不同于具体提供的和上面描述实施例都是显而易见的。例如，所说明的*FIFO*缓冲器65利用*FIFO*寄存器72的两部分，提供具有两种数据宽度的*FIFO*缓冲器。但是，在其他实施例中，*FIFO*缓冲器65可以适合具有三个或多个部分，提供具有三种或多种数据宽度的*FIFO*缓冲器。从而，试图以后附的权利要求书覆盖落入本发明的真正精神与范围的所有改进。



24 地址
控制
24 数据
外部地址总线开关
38
外部数据总线开关
44
14
数据处理系统
20
Y存储器
34
X存储器
32
程序RAM和指令超高速缓冲存储器
30
YAB
XAB
PAB
DAB
56
57
58
59
DRAM和SRAM总线接口和指令超高速缓冲存储器控制
42
数据ALU
54
程序地址发生器
52
串行通信接口
28
3
6
6
增强型串行同步接口
26
程序控制单元
程序解码控制器
50
程序控制
6
DDB
YDB
XDB
PDB
GDB
61
62
63
64
60
主接口图2
24
50
程序中断控制器
48
定时器
22
3
地址产生单元/DMA控制器
36
内部数据总线开关
40
46

图1

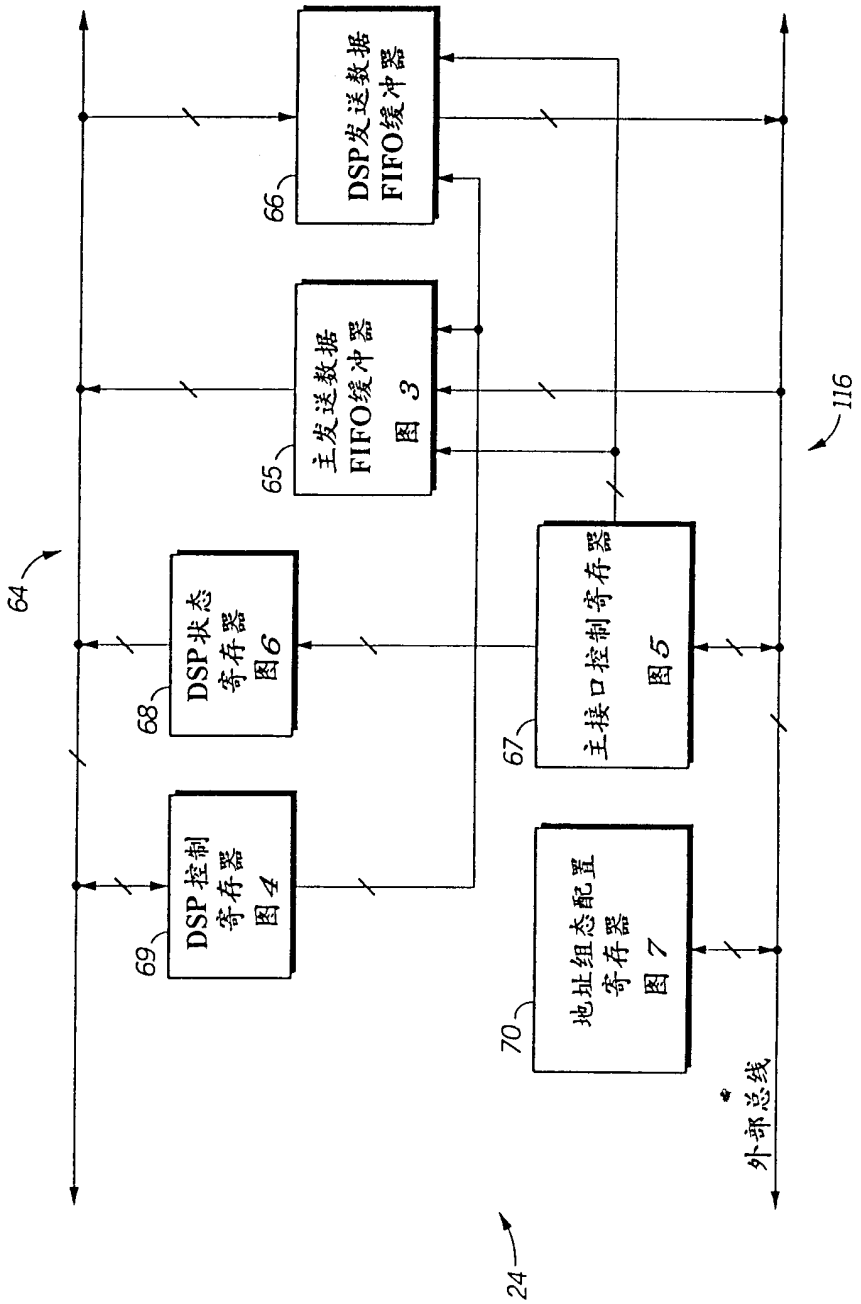

DSP发送数据FIFO缓冲器
66
主发送数据FIFO缓冲器 图 3
65
DSP状态寄存器 图6
68
DSP控制寄存器 图4
69
主接口控制寄存器 图5
67
地址组态配置寄存器 图 7
70
64
116
24
外部总线

图 2

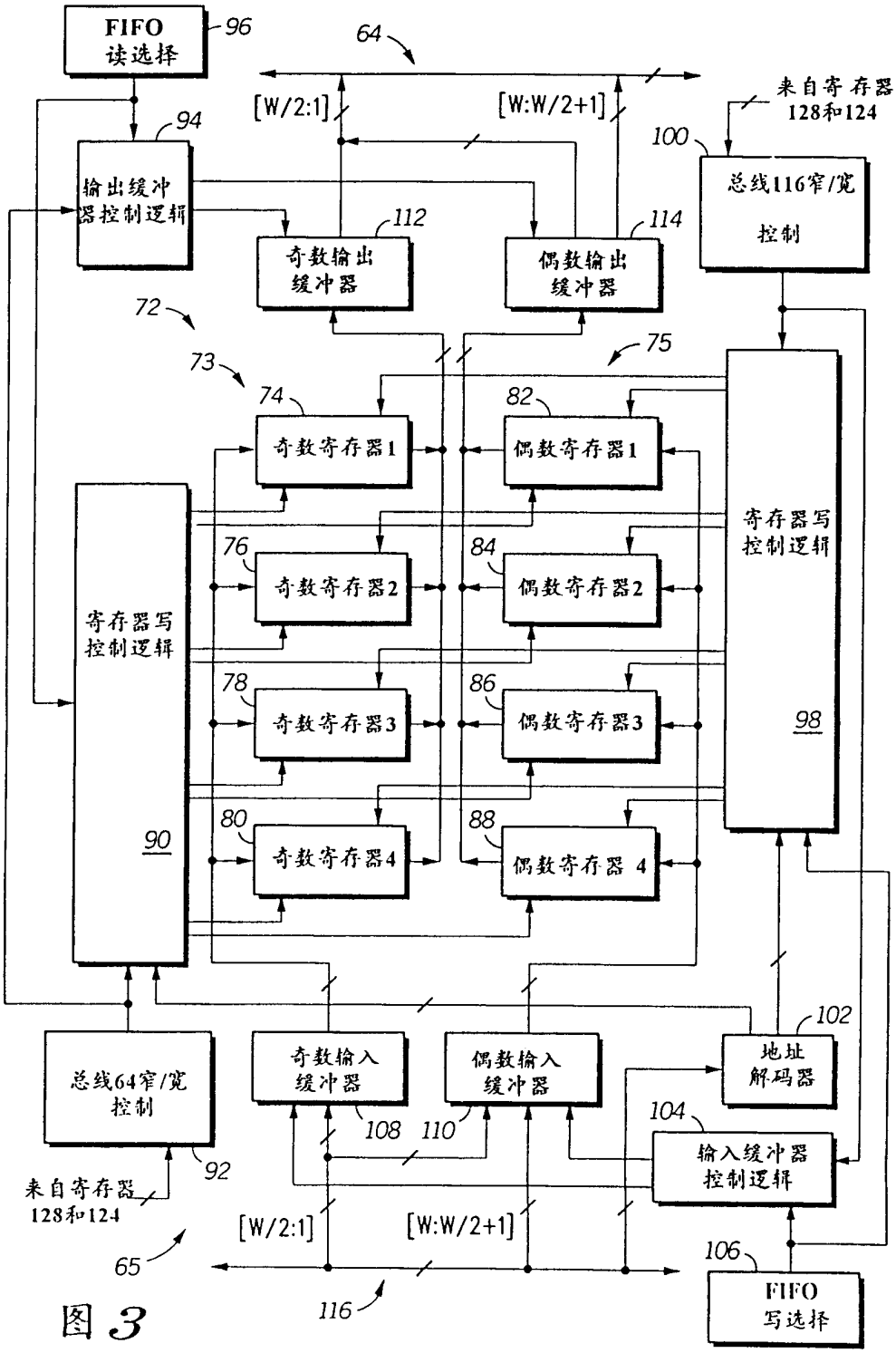

FIFO
读选择
96
64
[W/2:1]
[W:W/2+1]
来自寄存器
128和124
94
100
输出缓冲
器控制逻辑
总线116窄/宽
控制
112
114
奇数输出
缓冲器
偶数输出
缓冲器
72
73
75
74
82
奇数寄存器1
偶数寄存器1
寄存器写
控制逻辑
76
84
奇数寄存器2
偶数寄存器2
寄存器写
控制逻辑
98
78
86
奇数寄存器3
偶数寄存器3
80
88
90
奇数寄存器4
偶数寄存器 4
102
总线64窄/宽
控制
奇数输入
缓冲器
偶数输入
缓冲器
地址
解码器
104
108
110
输入缓冲器
控制逻辑
92
来自寄存器
128和124
[W/2:1]
[W:W/2+1]
106
65
116
FIFO
写选择

图 3

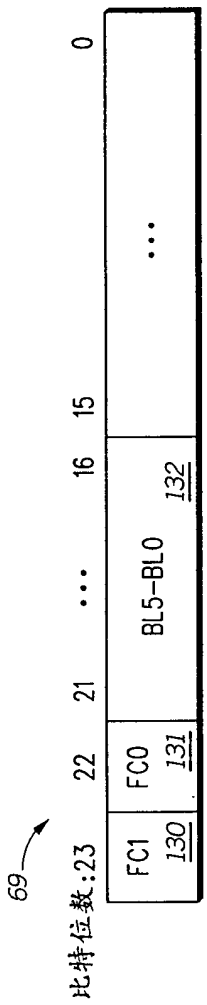

69
比特位数:23
22
21
16
15
0
FC1
130
FC0
131
BL5-BL0
132
...

图4

67
比特位数:31
15
14
13
12
11
10
5
4
3
0
...
135
HRF1
136
HRF0
137
HTF1
138
HTF0
139
...
HF2
140
HF1
141
HF0
142
...

图5

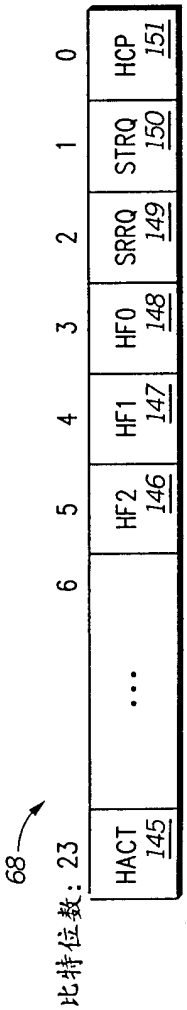

68
比特位数: 23
6
5
4
3
2
1
0
HACT
145
...
HF2
146
HF1
147
HF0
148
SRRQ
149
STRQ
150
HCP
151

图6

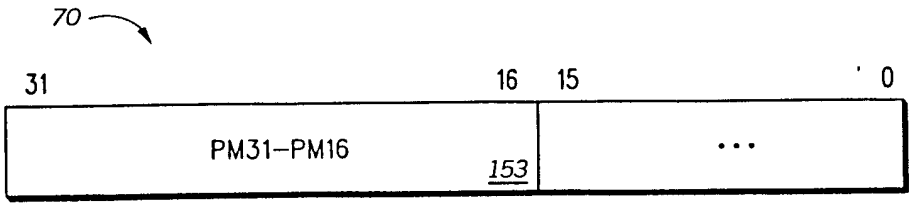

70
31
16
15
0
PM31–PM16
153
...

图7

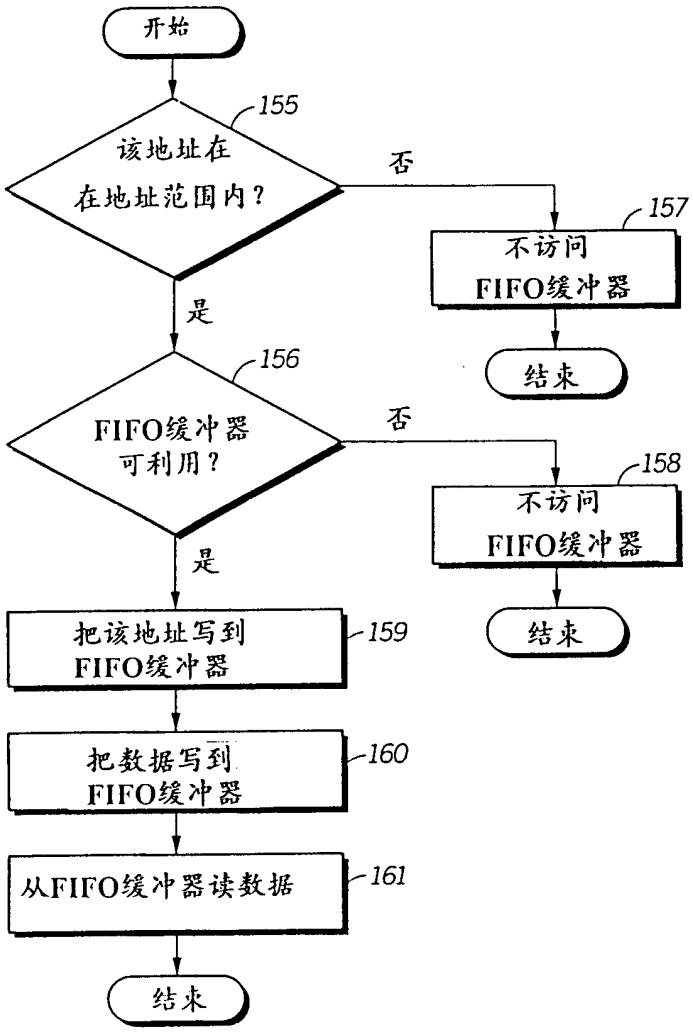

开始
155
该地址在
在地址范围内？
否
157
不访问
FIFO缓冲器
结束
是
156
FIFO缓冲器
可利用？
否
158
不访问
FIFO缓冲器
结束
是
把该地址写到
FIFO缓冲器
159
把数据写到
FIFO缓冲器
160
从FIFO缓冲器读数据
161
结束

图8