[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04N 5/04 (2006.01)
H04N 5/14 (2006.01)
G09G 5/00 (2006.01)

[12] 发明专利说明书

专利号 ZL 02818425.4

[45] 授权公告日 2008 年 2 月 20 日

[11] 授权公告号 CN 100370803C

[22] 申请日 2002.9.19 [21] 申请号 02818425.4

[30] 优先权

[32] 2001.9.19 [33] US [31] 60/323,238

[86] 国际申请 PCT/US2002/029808 2002.9.19

[87] 国际公布 WO2003/025900 英 2003.3.27

[85] 进入国家阶段日期 2004.3.19

[73] 专利权人 汤姆森许可公司

地址 法国布洛涅

[72] 发明人 P·D·菲利曼 M·D·克诺克斯

D·L·辛普森

[56] 参考文献

JP1188826A 1999.3.30

US5608425A 1997.3.4

US5537563A 1996.7.16

US5774178A 1998.6.30

审查员 王 超

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 程天正 王 勇

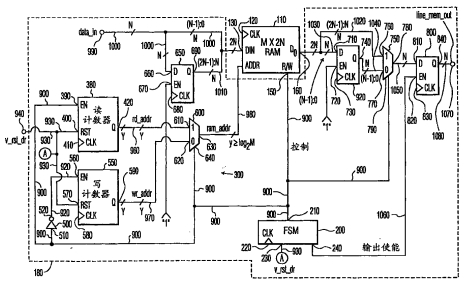
权利要求书 4 页 说明书 9 页 附图 4 页

[54] 发明名称

用于延时视频行数据的设备及其方法

[57] 摘要

提供了用于延时在发送设备与接收设备之间的视频行数据的设备(100)。该设备包括单端口随机存取存储器(“RAM”)(110)和处理装置，该处理装置包括被耦合到 RAM(110)的第一存储装置(700)和被耦合到 RAM(110)的第二存储装置(650)。



1. 一种用于延时在发送装置与接收装置之间的视频行数据的设备（100），该视频行数据包括相应于先前的视频行的数据和相应于现在的视频行的数据，相应于先前的视频行的数据包括第一数据部分和第二数据部分，以及相应于现在的视频行的数据包括第三数据部分和第四数据部分，该设备（100）包括：

单端口随机存取存储器RAM（110），它包括多个存储单元；以及

处理装置（180），它包括被耦合到 RAM（110）的第一存储装置（700）、状态控制器（200）、和被耦合到RAM（110）的第二存储装置（650），该处理装置（180）被配置成

从 RAM（110）的一个存储单元读出第一数据部分和第二数据部分，

把来自 RAM（110）的第一数据部分输出到接收装置，

把第二数据部分存储到第一存储装置（700），

把来自发送装置的第三数据部分存储到第二存储装置（650），

把来自第二存储装置（650）的第三数据部分和来自发送装置的第四数据部分写入到RAM（110）的一个存储单元，以及

把来自第一存储装置（700）的第二数据部分输出到接收装置；

其中该状态控制器（200）控制该RAM（110）的读和写状态。

2. 权利要求1的设备（100），其中：

该处理装置（180）还被配置成同时

从RAM(110)的一个存储单元读出第一数据部分和第二数据部分，

把第一数据部分输出到接收装置，

把第二数据部分存储到第一存储装置（700），以及

把第三数据部分存储到第二存储装置（650），

以及该处理装置（180）还被配置成同时

把来自第二存储装置（650）的第三数据部分和来自发送装置的第四数据部分写入到RAM（110）的一个存储单元，以及

把来自第一存储装置（700）的第二数据部分输出到接收装置。

3. 权利要求2的设备（100），其中：

该处理装置（180）还被配置成

从RAM(110)的一个存储单元读出第一数据部分和第二数据部分，

把第一数据部分输出到接收装置，

把第二数据部分存储到第一存储装置（700），以及

把第三数据部分存储到第二存储装置（650），

以及该处理装置（180）还被配置成交替地

把来自第二存储装置（650）的第三数据部分和来自发送装置的第四数据部分写入到 RAM（110）的一个存储单元，以及

把来自第一存储装置（700）的第二数据部分输出到接收装置。

4. 权利要求 3 的设备（100），其中：

处理装置（180）还被配置成以相应于时钟信号的速率

从 RAM（110）的一个存储单元读出第一数据部分和第二数据部分，

把第一数据部分输出到接收装置，

把第二数据部分存储到第一存储装置（700），以及

把第三数据部分存储到第二存储装置（650），

以及处理装置（180）还被配置成，以相同的速率，

把来自第二存储装置（650）的第三数据部分和来自发送装置的第四数据部分写入到 RAM（110）的一个存储单元，以及

把来自第一存储装置（700）的第二数据部分输出到接收装置。

5. 权利要求 4 的设备（100），该其中处理装置（180）还包括用于顺序地递增第一地址、用于顺序地递增第二地址、以及用于交替地提供第一地址和第二地址到 RAM（110）的计数装置（300）。

6. 权利要求 1 的设备（100），其中该状态控制器（200）是一个有限状态机 FSM。

7. 权利要求 1 的设备（100），其中：

第一存储装置（700）包括第一数目的第一 D 触发器，

第二存储装置（650）包括第二数目的第二 D 触发器，以及

该第一 D 触发器的第一数目等于该第二 D 触发器的第二数目。

8. 权利要求 1 的设备（100），其中 RAM（110）是动态 RAM。

9. 一种用于延时在发送装置与接收装置之间的视频行数据的方法（1250），该视频行数据包括相应于先前的视频行的数据和相应于现在的视频行的数据，该相应于先前的视频行的数据包括第一数据部分和第二数据部分，以及该相应于现在的视频行的数据包括第三数据部分和第四数据部分，该方法（1250）包括以下步骤：

从单端口随机存取存储器 RAM（110）的一个存储单元读出（1320）第一数据部分和第二数据部分；

把第一数据部分输出（1330）到接收装置；

把第二数据部分存储（1340）到第一存储装置（700）；

把第三数据部分存储（1350）到第二存储装置（650）；

把来自第二存储装置（650）的第三数据部分和来自发送装置的第四数据部分写入（1370）到 RAM（1I0）的存储单元；以及

把来自第一存储装置（700）的第二数据部分输出（1380）到接收装置；

其中该读和写步骤发生在不同时间。

10. 权利要求 9 的方法（1250），其中：

读出（1320）第一数据部分和第二数据部分、输出（1330）第一数据部分、存储（1340）第二数据部分、和存储（1350）第三数据部分的各所述步骤是同时执行的，以及

写入（1370）第三数据部分和第四数据部分以及输出（1380）第二数据部分的各所述步骤是同时执行的。

11. 权利要求 10 的方法（1250），其中：

（a）读出（1320）第一数据部分和第二数据部分、输出（1330）第一数据部分、存储（1340）第二数据部分、和存储（1350）第三数据部分的各所述步骤是与（b）写入（1370）第三数据部分和第四数据部分以及输出（1380）第二数据部分的各所述步骤交替地执行的。

12. 权利要求 11 的方法（1250），其中：

（a）读出（1320）第一数据部分和第二数据部分、输出（1330）第一数据部分、存储（1340）第二数据部分、和存储（1350）第三数据部分的各所述步骤是以相应于时钟信号的速率执行的，以及（b）写入（1370）第三数据部分和第四数据部分以及输出（1380）第二数据部分的步骤是以相同的速率执行的。

13. 权利要求 12 的方法（1250），还包括以下步骤：

顺序地递增（1354）第一地址；

顺序地递增（1384）第二地址；以及

把第一地址和第二地址交替地提供（1290，1300）到 RAM（110）。

14. 权利要求 9 的方法（1250），其中：

存储（1350）第三数据部分的该步骤包括把第三数据部分存储在第一数目的第一 D 触发器中，以及

存储（1340）第二数据部分的该步骤包括把第二数据部分存储在第二数目的第二 D 触发器中，

其中该第一 D 触发器的第一数目与该第二 D 触发器的第二数目相等。

15. 权利要求 9 的方法（1250），其中该单端口随机存取存储器是动态 RAM。

用于延时视频行数据的设备及其方法

优先权

本专利申请要求2001年9月19日提交的、题目为“使用单端口存储器的视频延时行”的、美国临时专利申请No.60/323,238的利益，该专利申请在此引用，以供参考。

发明领域

本发明涉及数字信号处理领域。

发明背景

传统的视频信号包含一系列帧。每个帧包含一系列行，以及每行包含多个象素。需要视频行延时（或“视频延时行”）来执行垂直格式变换和图象信号改进。其他数字信号处理应用（诸如音频滤波）和其他计算机有关的应用也需要把数字数据延时确定的数目的时钟周期。典型的行延时是通过使用先进先出队列（“FIFO”）构建的，行数据被馈送到FIFO的输入端，以及以取决于所需要的延时量和FIFO大小的速率经时钟控制传送到FIFO输出端。

典型地，当数据量足够大，使得随机存取存储器（“RAM”）实施方案比起替换方案（触发器或锁存器）更实际时，RAM被使用于FIFO。对于在互相无关的时间写数据到RAM和从RAM读出数据的系统，典型地使用双端口RAM。双端口RAM具有独立的读和写端口。特别是，双端口允许数据被写入到一个RAM地址以及同时从另一个地址读数据，这易于实行延时设计。然而，双端口RAM的缺点是它们的硅面积。双端口RAM可以比起可比较的单端口RAM大100%。另外，双端口RAM是非常昂贵的。

本发明针对克服这个问题。

发明概要

用于延时在发送设备与接收设备之间的视频行数据的设备包括单端口随机存取存储器（“RAM”）和处理装置。处理装置被配置成从RAM的存储单元读出该数据的第一多个部分之一和该数据的第二多个部分之一（相应于先前的视频行）、把该数据的第一多个部分之一输出到接收设备、把该数据的第二多个部分之一存储到第一存储装置、把该

数据的第三多个部分之一（相应于现在的视频行）存储到第二存储装置、把来自第二存储装置的该数据的第三多个部分之一和来自发送设备的该数据的第四多个部分之一（也相应于现在的视频行）写入到该存储单元、以及把来自第一存储装置的该数据的第二多个部分之一输出到该接收设备。

附图简述

图上:

图 1 是按照本发明的示例性数字行延时的方框图;

图 2 是图 1 的示例性数字行延时的有限状态机（"FSM"）的示例性运行的状态图;

图 3 是图 1 的数字行延时的示例性运行的流程图; 以及

图 4 是图 1 的数字行延时的示例性运行的时序图。

优选实施例详细描述

通过作为例子给出的以下的说明，将更加明白本发明的特征和优点。

图 1 是按照本发明的示例性数字行延时 100 的方框图。行延时 100 包括 M 个字、每字 2N 比特的单端口 RAM 110。当用在这里时，术语"字节"相应于 N 比特的数据，以及术语"字"相应于 2N 比特的数据。在传统的视频应用中，一行视频数据典型地由 1,920 个象素组成。因此，对于视频应用，M 可以是 960 和 N 可以是 8，因此 RAM 110 可以是 960x16。应当看到，这样的配置假定: RAM 110 可保持每个象素 8 比特的视频数据的一行的有用成分（即，960 字乘以每字 2 象素=1,920 象素=1 行）。然而，应当指出，在替换实施例中，M 和 N 可以是任何其他适当的数值，这样，RAM 110 被配置成适合于特定应用的存储容量。RAM 110 包括时钟输入端 120、1 个字或 2N 比特宽（N 是以上讨论的）数据输入端 130、Y 比特宽地址输入端 140（其中 M 是以上讨论的，以及 $Y \geq 1\log_2 M$, 这样接收的地址输入字的范围可以足够覆盖 RAM 110 的 M 个地址）、读/写（"R/W"）控制输入端 150、和 1 个字或 2N 比特宽的数据输出端 160。而且，在这里描述的示例性实施例中，RAM 110 是静态 RAM（"SRAM"）。然而，应当看到，在替换实施例中，RAM 110 可以是动态 RAM（"DRAM"）或任何其他适用的类型的单端口 RAM。

行延时 100 还包括处理装置 180。处理装置 180 包括有限状态机

（“FSM”）200。FSM 被配置成调节或控制行延时 100 的运行，正如下面更详细地讨论的。FSM 200 包括控制输出端 210、时钟输入端 220、复位输入端 230、和使能输出端 240。

处理装置 180 还包括地址计数装置 300。计数装置 300 被配置成提供指定 RAM 100 的地址的 RAM 地址号，在该地址行数据被交替地读和写，正如下面进一步讨论的。计数装置 300 包括二进制计数器 380，它具有使能输入端 390、复位输入端 400、时钟输入端 410、和 Y 比特宽（Y 是以上讨论的）输出端 420。计数装置 300 还包括反相器 500，它具有输入端 510 和输出端 520。计数装置 300 还包括二进制计数器 550，它具有使能输入端 560、复位输入端 570、时钟输入端 580、和 Y 比特宽输出端 590。另外，计数装置 300 包括数据开关或复接器 600，它具有 Y 比特宽数据输入端 610、Y 比特宽数据输入端 620、Y 比特宽数据输出端 630、和控制输入端 640。复接器 600 被配置成当在它的控制输入端 640 处接收逻辑 1 时，把在它的输入端 610 处接收的数据传送到它的输出端 630，否则，把在它的输入端 620 处接收的数据传送到它的输出端 630。

处理装置 180 还包括 N 个（N 是以上讨论的）D 触发器 650 的组或库，它们具有一组 N 个各个 D 输入端 660、一组 N 个各个使能输入端 670、一组 N 个各个时钟输入端 680、和一组 N 个各个 Q 输出端 690。处理装置 180 还包括 N 个 D 触发器 700 的组或库，它们具有一组 N 个各个 D 输入端 710、一组 N 个各个使能输入端 720、一组 N 个各个时钟输入端 730、和一组 N 个各个 Q 输出端 740。使能输入端 670 和使能输入端 720 都被耦合到逻辑 1。

处理装置 180 还包括数据开关或复接器 750，它具有 N 比特宽数据输入端 760、N 比特宽数据输入端 770、N 比特宽数据输出端 780、和控制输入端 790。复接器 750 被配置成当在它的控制输入端 790 处接收逻辑 1 时，把在它的输入端 760 处接收的数据传送到它的输出端 780，否则，把在它的输入端 770 处接收的数据传送到它的输出端 780。另外，处理装置 180 包括 N 个 D 触发器 800 的组或库，它们具有一组 N 个各个 D 输入端 810、一组 N 个各个使能输入端 820、一组 N 个各个时钟输入端 830、和一组 N 个各个 Q 输出端 840。

处理装置 180 还包括导线 900，它把控制输入端 210 耦合到 RAM 110

的 R/W 控制输入端 150、计数器 380 的使能输入端 390、反相器 500 的输入端 510、复接器 600 的控制输入端 640、和复接器 750 的控制输入端 790。处理装置 180 还包括导线 920，它把反相器 500 的输出端 520 耦合到计数器 550 的使能输入端 560。处理装置 180 还包括导线 930，它把计数器 380 的复位输入端 400 耦合到计数器 550 的复位输入端 570 和 FSM 200 的复位输入端 230。处理装置 180 还包括垂直复位输入端 940。导线 930 也把复位输入端 400、复位输入端 570、和复位输入端 230 耦合到垂直复位输入端 940。

处理装置 180 还包括 Y 导线 960 的组或库（Y 是以上讨论的），它们把计数器 380 的每个各个 Q 输出端 420 耦合到复接器 600 的输入端 610 的各个比特。处理装置 180 还包括 Y 导线 970 的组或库，它们把计数器 550 的每个各个 Q 输出端 590 耦合到复接器 600 的输入端 620 的各个比特。处理装置 180 还包括导线 980 的组或库，它们把复接器 600 的每个各个输出端 630 耦合到 RAM 110 的地址输入端 140 的各个比特。

处理装置 180 还包括 N 比特宽（N 是以上讨论的）数据入端口 990 和 N 导线 1000 的组或库，它们把数据入端口 990 的比特耦合到 RAM 110 的数据输入端 130 的各个 (N-1):0 比特。导线 1000 也把数据入端口 990 的比特耦合到 D 触发器 650 的各个 D 输入端 660。处理装置 180 还包括 N 导线 1010 的组或库，它们把 D 触发器 650 的 Q 输出端 690 耦合到 RAM 110 的数据输入端 130 的各个 (2N-1):N 比特。

处理装置 180 还包括 N 导线 1020 的组或库（N 是以上讨论的），它们把来自 RAM 110 的数据输出端 160 的各个输出 (2N-1):N 耦合到复接器 750 的输入端 760。处理装置 180 还包括 N 导线 1030 的组或库，它们把来自 RAM 110 的数据输出端 160 的各个输出 (N-1):0 耦合到 D 触发器 700 的 D 输入端 710。处理装置 180 还包括 N 导线 1040 的组或库，它们把 D 触发器 700 的 Q 输出端 740 耦合到复接器 750 的各个输入端 770。而且，处理装置 180 包括 N 导线 1050 的组或库，它们把复接器 750 的 N 比特宽输出端 780 耦合到 D 触发器 800 的各个 D 输入端 810。处理装置 180 还包括导线 1060，它把 FSM 200 的使能输出端 240 耦合到 D 触发器 800 的所有的使能输入端 820。另外，处理装置 180 还包括 N 比特宽行存储器输出端 1070 和 N 导线 1080 的组或库，它们

把 D 触发器 800 的 Q 输出端 840 耦行存储器输出端 1070。

图 2 是图 1 的示例性数字行延时 100 的 FSM 的示例性运行 1200 的状态图。在该示例性实施例中，FSM 200 只有两个状态：读状态 1210 和写状态 1220。在运行时，示例性 FSM 200 在它的控制输出端 210 处提供 CONTROL(控制)信号，以及在它的使能输出端 240 处提供 ENABLE（使能）信号。在读状态 1210，FSM 使得 CONTROL 信号成为逻辑 1，以及使得 ENABLE 信号变成或保持逻辑 1。在写状态 1220，FSM 使得 CONTROL 信号变成逻辑 0，以及使得 ENABLE 信号变成或保持逻辑 1。而且，FSM 在检测到通过垂直复位输入端 940 和导线 930 从外部设备（未示出）接收的惯用的视频垂直复位信号（“V_RST_DR”）的逻辑 1 状态或脉冲后从写状态 1200 重新启动或复位。虽然示例性实施例的一个好处是 FSM 200 只有两个状态，但应当看到，在替换实施例中，FSM 200 可以执行附加运行，因此，可以具有比起图 2 所示的更多的状态，和/或具有比起图 1 所示的更多的输入端或输出端。

图 3 是图 1 的数字行延时 100 的示例性运行 1250 的流程图。在运行时，外部系统时钟（未示出）以熟知的方式提供惯用的同步时钟信号（“CLOCK”）到行延时 100 的各个部件（经过时钟输入端 120、时钟输入端 220、时钟输入端 410、时钟输入端 580、时钟输入端 680、时钟输入端 730、时钟输入端 830 等，－见图 1）。应当看到，当 CLOCK 信号同步地驱动行延时 100 的各个部件时，行延时 100 在 CLOCK 信号的每个周期或脉冲内或多或少同时地执行运行的一个迭代，因此，流程图 1250 仅仅是从 CLOCK 周期到 CLOCK 周期的运行的概念性特性的例子，并不意味着把本发明限制于在 CLOCK 周期的过程或范围期间特定的运行的序列或次序。

在步骤 1260，行延时 100 以熟知的方式执行适当的电源接通初始化运行。在步骤 1260 运行后，行延时 100 进到步骤 1270。在步骤 1270，行延时 100 确定外部设备（未示出）是否调用了处理复位。在示例性实施例中，行延时 100 根据经过垂直复位输入端 490 从外部设备（未示出）接收的惯用的视频垂直复位信号，V_RST_DR，作出这个判决。如果复位被调用，则行延时 100 进到步骤 1280；否则行延时 100 进到步骤 1290。

在步骤 1280，行延时 100 执行适当的处理复位运行。步骤 1280

的运行包括 FSM 200 复位到它的写状态 1220（见以上讨论的图 2）。步骤 1280 的运行还包括行延时 100 以熟知的方式使得计数器 380 复位（即，所有的 Q 输出端 420 变成 0）和使得计数器 550 复位（即，所有的 Q 输出端 590 变成 0）。在步骤 1280 的运行后，行延时 100 循环回到步骤 1270。

在步骤 1290，行延时 100 从外部系统时钟（未示出）接收 CLOCK 信号的脉冲或过渡。在步骤 1290 的运行后，行延时 100 进到步骤 1300。在步骤 1300，行延时 100 确定 CLOCK 信号的现在的脉冲是奇数号脉冲（即，自从上一次处理复位以来接收的第一、第三、或第五，...等等脉冲）还是偶数号脉冲（即，自从上一次处理复位以来接收的第二、第四、或第六，...等等脉冲）。在示例性实施例中，这个判决由只有两个状态的 FSM 200 实行（见图 2）。FSM 200 在它接收到复位后的第一个时钟脉冲后假设读状态 1210，在它接收到复位后的第二个时钟脉冲后假设写状态 1220，等等，因为它分别对于奇数时钟脉冲和偶数时钟脉冲在读状态 1210 与写状态 1220 之间进行交替。当行延时 100 确定 CLOCK 信号提供奇数号脉冲时，行延时 100 进到步骤 1320-1354；否则行延时 100 进到步骤 1370-1384。

在步骤 1320，行延时 100 从 RAM 110 的地址号 RD_ADDR 读出数据的字（即，两个字节）。在该示例性实施例中，这是通过 RAM 110 在它的地址输入端 140 处接收 RD_ADDR 号、通过 RAM 110 在它的 R/W 控制输入端 150 处接收 CONTROL 信号、通过复接器 750 在输入端 760 处接收字的比特(2N-1):N、和通过 D 触发器 700 在 D 输入端 710 处接收字的比特(N-1):0，而实行的。在步骤 1320 的运行后，行延时 100 进到步骤 1330。

在步骤 1330，行延时 100 把在步骤 1320 读出的字的比特(2N-1):N(即，“高”字节)通过复接器 750 传送到 D 触发器 800。应当看到，因为 FSM200 处在读状态 1210，所以复接器 750 传送高字节（见以上讨论的步骤 1300），因此，在控制输入端 790 处的 CONTROL 信号对于这些运行是逻辑 1。D 触发器 800 缓存高字节，之后把它经过 Q 输出端 840 传送到行存储器输出端 1070。然而，应当指出，通过 D 触发器 800 的这个缓存仅仅是示例性的，对于本发明并不重要。在其中 D 触发器 800 被省略的替换实施例中，行延时 100 把高字节直接传送到行存储器

输出端 1070。在步骤 1330 的运行后，行延时 100 进到步骤 1340。

在步骤 1340，行延时 100 把在步骤 1320 读出的字的比特(N-1):0(即，“低”字节)存储在 D 触发器 700。于是，应当看到，D 触发器 700 提供“读缓存器”，它存储或保持低字节，而高的字节进到复接器 750。在步骤 1340 的运行后，行延时 100 进到步骤 1350。

在步骤 1350，行延时 100 也存储由外部设备（未示出）提供的进入的行数据的现在的字节。行延时 100 在数据入端口 990 接收数据的这个字节，以及把它存储在 D 触发器 650。应当看到，虽然这个字节被包括在出现在 RAM 110 的数据输入端 130 处的字中，但 RAM 110 在步骤 1350 运行期间丢弃在数据输入端 130 处的字，因为 RAM 110 在它的 R/W 控制输入端处接收到 CONTROL 信号是逻辑 1（FSM 200 处在读状态 1210），这命令进行读运行。在步骤 1350 的运行后，行延时 100 进到步骤 1354。

在步骤 1354，行延时 100 使读地址（“RD_ADDR”）递增。RD_ADDR 是由计数器 380 的 Q 输出端 420 表示的二进制数。在该示例性实施例中，这是通过 FSM 200 在它的控制输出端 210 提供的 CONTROL 信号的状态实行的。当 FSM 200 处在读状态 1210 时，CONTROL 信号是逻辑 1。当计数器 380 在使能输入端 390 处接收逻辑 1 的 CONTROL 信号时，计数器 380 被使能。计数器 380 也接收现在的 CLOCK 脉冲，随之使 Q 输出 420 递增。另外，响应于 CONTROL 信号的逻辑 1 状态，复接器 600 把 RD_ADDR 号传送到它的输出端 630。在步骤 1354 的运行后，行延时 100 循环回到步骤 1270。

在步骤 1370，行延时 100 把由先前存储的数据的字节（见以上讨论的步骤 1350）和来自外部设备的进入的行数据的新的现在的字节组成的字，分别作为比特(2N-1):N 和(N-1):0 写入到 RAM 110 的地址号 WR_ADDR。行延时 100 从 D 触发器 650 的 Q 输出端 690 检索先前存储的数据的字节。RAM 110 把字写入到地址号 WR_ADDR，因为在这些运行期间 FSM 200 处在写状态 1220（见在以上讨论的步骤 1300），这使得 RAM 110 在它的 RW 控制输入端 150 处接收到 CONTROL 信号是逻辑 0，它命令进行写运行，以及因为复接器 600 把 WR_ADDR 号提供到地址输入端 140。在步骤 1370 的运行后，行延时 100 进到步骤 1380。

在步骤 1380，行延时 100 把在步骤 1320 从 D 触发器 700 的 Q 输

出端 740 读出的字的比特(N-1):0(即，“低”字节)通过复接器 750 传送到 D 触发器 800。应当看到，因为 FSM 处在写状态 1220，所以复接器 750 传送来自 Q 输出端 740 的低字节(见以上讨论的步骤 1300)，因此，在控制输入端 790 处的 CONTROL 信号对于这些运行是逻辑 0。像步骤 1330 那样，D 触发器 800 缓存低字节，之后把它经过 Q 输出端 840 传送到行存储器输出端 1070。再一次地，通过 D 触发器 800 的这个缓存仅仅是示例的，对于本发明并不重要。在其中 D 触发器 800 被省略的替换实施例中，行延时 100 把低字节直接传送到行存储器输出端 1070。在步骤 1380 的运行后，行延时 100 进到步骤 1384。

在步骤 1384，行延时 100 使写地址(“WR_ADDR”)递增。WR_ADDR 是由计数器 550 的 Q 输出端 590 表示的二进制数。在示例性实施例中，这是通过 FSM 200 在它的控制输出端 210 提供的 CONTROL 信号的状态实行的。当 FSM 200 处在写状态 1220 时，CONTROL 信号是逻辑 0。当逻辑 0 的 CONTROL 信号被反相器 500 反相和最终得到的逻辑 1 在计数器 550 的使能输入端 560 处被接收时，计数器 550 被使能。计数器 550 也接收现在的 CLOCK 脉冲，并随之使 Q 输出 590 递增。另外，响应于 CONTROL 信号的逻辑 0 状态，复接器 600 把 WR_ADDR 号传送到它的输出端 630。在步骤 1384 的运行后，行延时 100 循环回到步骤 1270。

图 4 是图 1 的数字行延时 100 的示例性运行 1450 的时间图。RD_ADDR 从 0 开始，在奇数号 CLOCK 周期递增，范围从 0 到 M(M 是在以上讨论的)，复位到 0，以及继续重复这个序列。WR_ADDR 从 0 开始，在偶数号 CLOCK 周期递增，范围从 0 到 M，复位到 0，以及继续重复这个序列。RAM_ADDR 从 0 开始，在奇数号 CLOCK 周期递增，范围从 0 到 M,复位到 0,以及继续重复这个序列。应当看到,行延时 100 在 CONTROL 信号处在逻辑 1 的每个 CLOCK 周期(FSM 200 处在读状态 1210)，执行“读循环”(以上讨论的步骤 1320-1354),以及行延时 100 在 CONTROL 信号处在逻辑 0 的每个 CLOCK 周期(FSM 200 处在写状态 1220)，执行“写循环”(以上讨论的步骤 1370-1384)。因此，示例性行延时 100 在读和写运行之间时间复用单端口 RAM 110。另外，示例性 FSM 200 仅仅用两个状态调节行延时 100 的运行。应当看到，在接通电源后由行延时 100 从 RAM 110 首先全部读出或输出数据(即，在接通电源后第一组 CLOCK 周期，在此期间或者 RD_ADDR 或者 WR_ADDR 不是 0)，

提供在接通电源时在 RAM 110 中任意出现的无论什麽数据。此后，因为示例性行延时 100 从 RAM 110 的 M 地址的每个地址从最后的或先前的行读出存储的数据，此后把来自新的或现在的行的数据写入到该地址，所以，示例性行延时 100 提供一行延时。

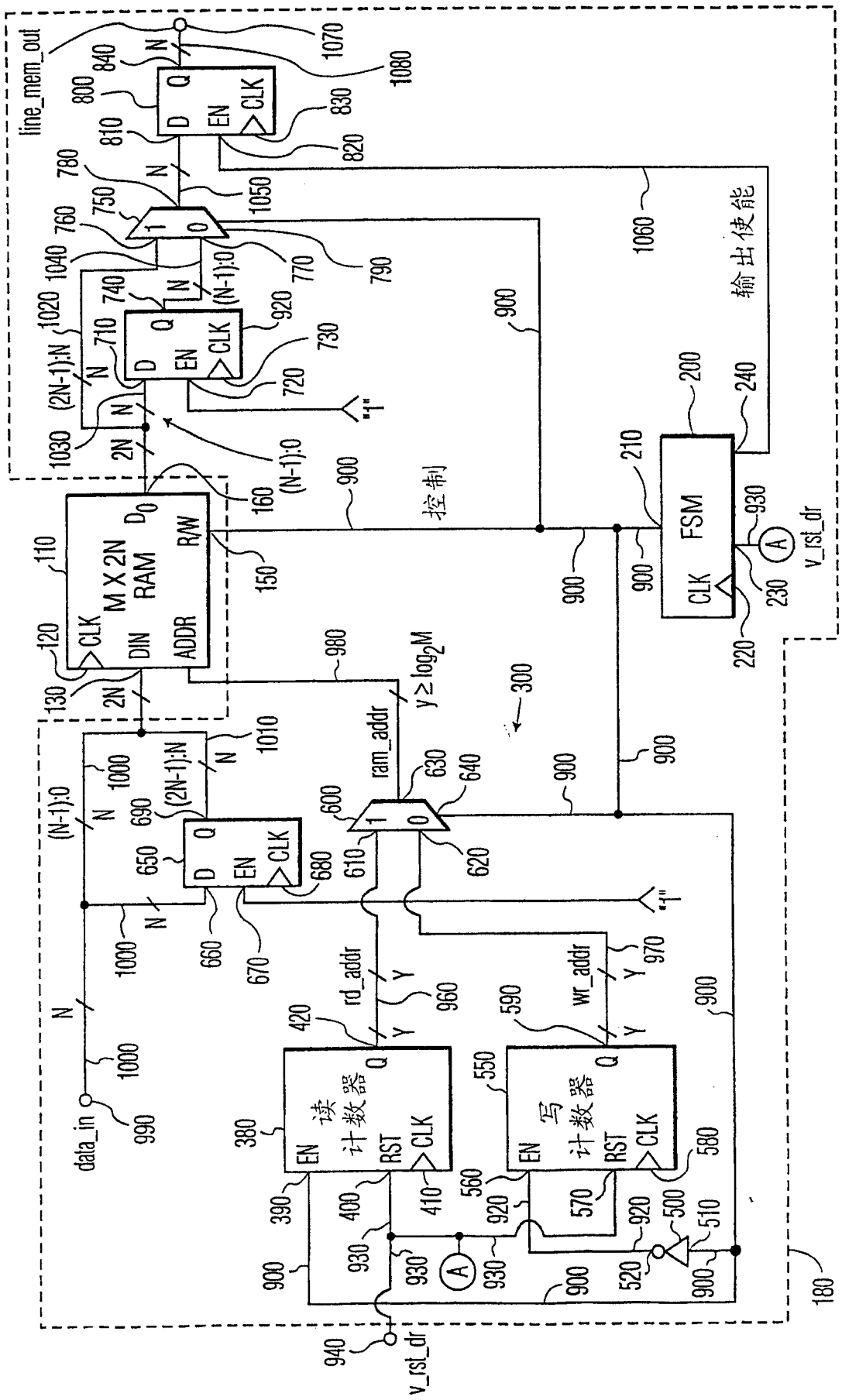

line_mem_out
M X 2N
RAM
CLK
DIN
ADDR
R/W
FSM
CLK
输出使能
控制
y ≥ log2M
ram_addr
rd_addr
wr_addr
data_in
读
计数器
写
计数器
EN
RST
CLK
v_rst_dr
"1"

图 1

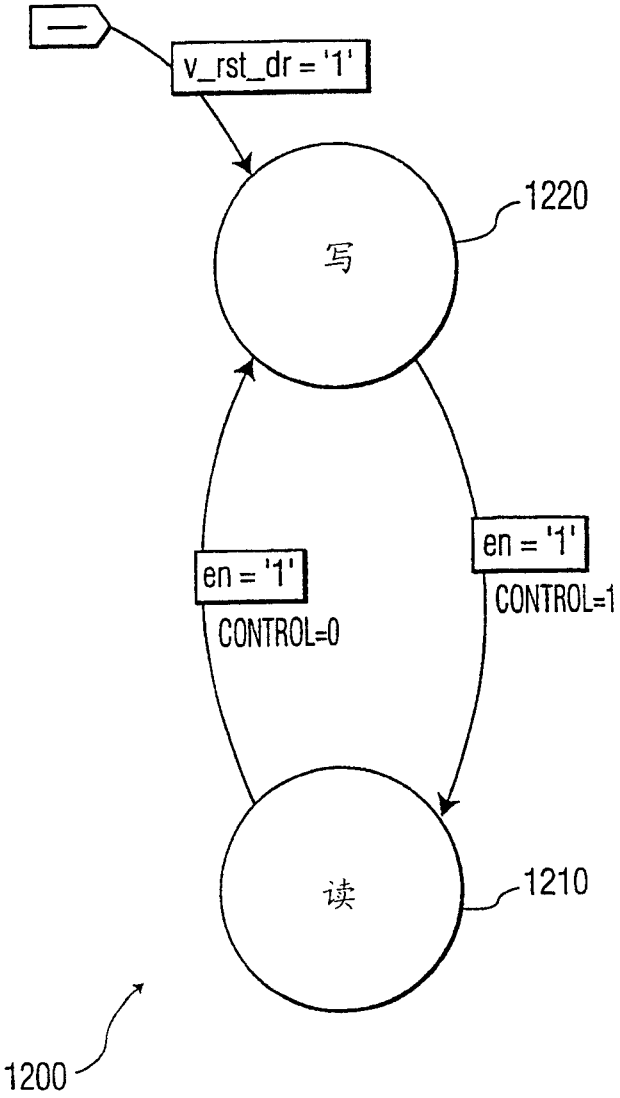


图 2

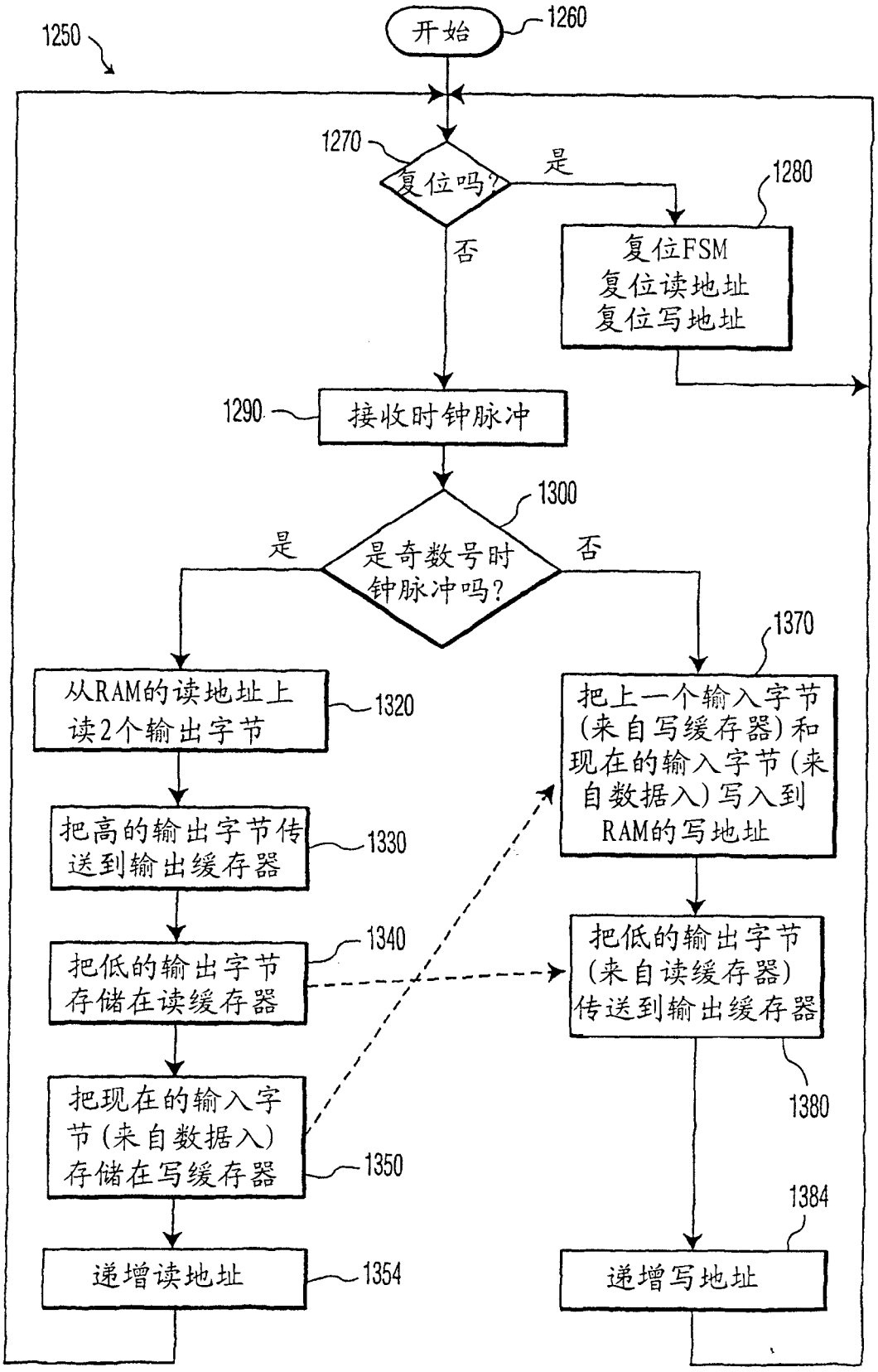

1250
开始
1260
1270
复位吗?
是
1280
复位FSM
复位读地址
复位写地址
否
1290
接收时钟脉冲
1300
是奇数号时钟脉冲吗?
是
否
1370
从RAM的读地址上读2个输出字节
1320
把上一个输入字节(来自写缓存器)和现在的输入字节(来自数据入)写入到RAM的写地址
把高的输出字节传送到输出缓存器
1330
把低的输出字节存储在读缓存器
1340
把低的输出字节(来自读缓存器)传送到输出缓存器
1380
把现在的输入字节(来自数据入)存储在写缓存器
1350
1384
递增读地址
1354
递增写地址

图 3

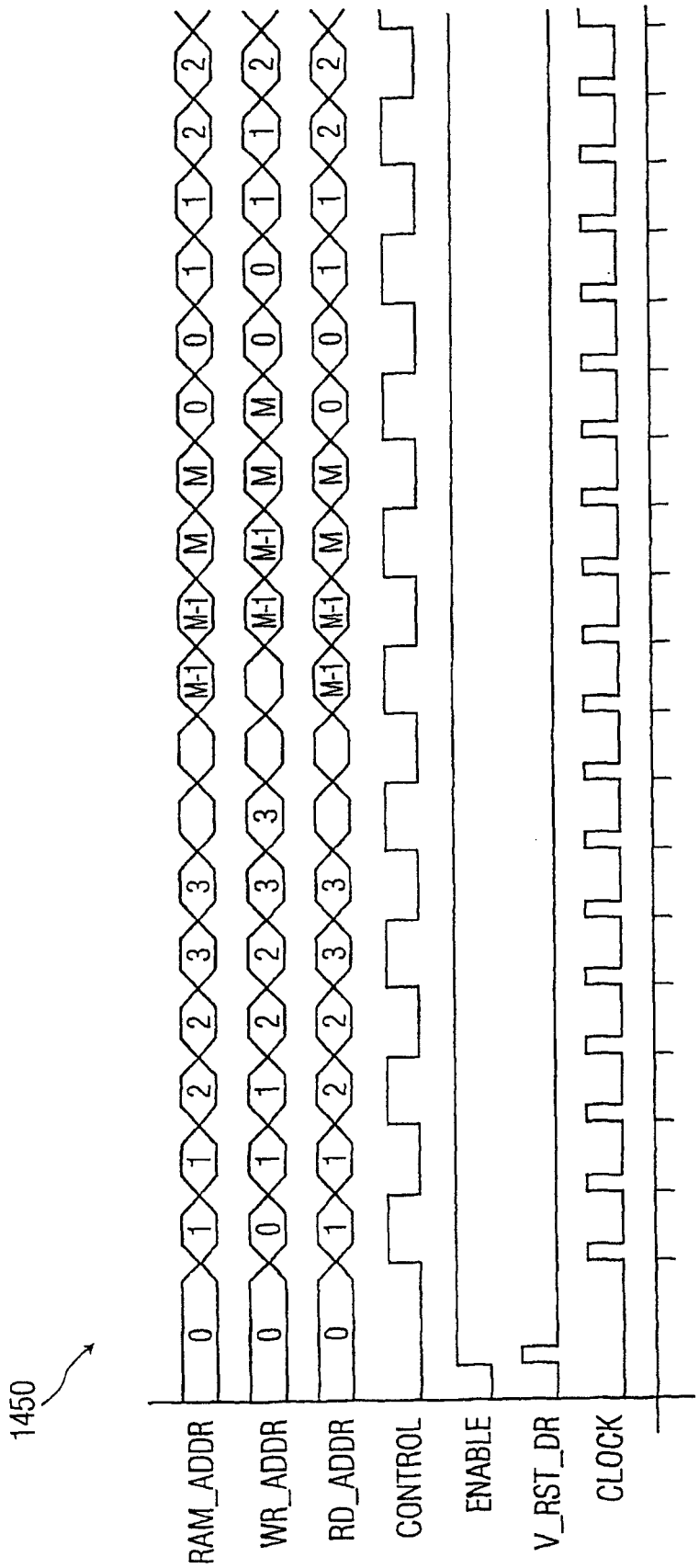

1450
RAM_ADDR
WR_ADDR
RD_ADDR
CONTROL
ENABLE
V_RST_DR
CLOCK
0
1
2
3
M-1
M

图 4