(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104620193 A
(43)申请公布日 2015.05.13

(21)申请号 201380046971.3
(22)申请日 2013.09.10
(30)优先权数据
61/698,906 2012.09.10 US
13/769,963 2013.02.19 US
(85)PCT国际申请进入国家阶段日
2015.03.10
(86)PCT国际申请的申请数据
PCT/US2013/059036 2013.09.10
(87)PCT国际申请的公布数据
WO2014/040065 EN 2014.03.13
(71)申请人 德克萨斯仪器股份有限公司
地址 美国德克萨斯州
(72)发明人 S·C·巴特玲 S·汉纳
(74)专利代理机构 北京纪凯知识产权代理有限公司 11245
代理人 赵蓉民
(51)Int.Cl.
G06F 1/26(2006.01)

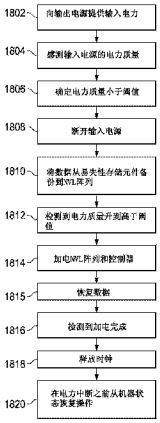
权利要求书3页 说明书24页 附图21页

(54)发明名称

在不一致供电期间的基于非易失性逻辑阵列的计算

(57)摘要

感测处理设备(1700)的输入电力质量。响应于不良电力质量的检测,断开输入电源,并且处理设备(1700)使用可用的存储的电荷,将它的机器状态备份在非易失性逻辑元件阵列(1710)中。当恢复电力时,将存储的机器状态从非易失性逻辑元件阵列(1710)恢复到易失性逻辑元件(1720),由此处理设备(1700)在电力丢失之前立即恢复它的该状态的处理,从而允许间歇性供电之间的无缝处理。



CN 104620193 A

1. 一种计算设备装置，其提供基于非易失性逻辑的计算，该装置包括：

处理设备，其包括：

多个非易失性逻辑元件阵列；

多个易失性存储元件；以及

至少一个非易失性逻辑控制器，其被配置为控制所述多个非易失性逻辑元件阵列，以存储由所述多个易失性存储元件表示的机器状态，并将存储的机器状态从所述多个非易失性逻辑元件阵列读出到所述多个易失性存储元件；

电压或电流检测器，其被配置为感测来自输入电源的电力质量；

电源管理控制器，其与所述电压或电流检测器通信以接收关于所述电力质量的信息，所述电源管理控制器被配置为与所述至少一个非易失性逻辑控制器通信以提供实现将所述机器状态存储至所述多个非易失性逻辑元件阵列和从所述多个非易失性逻辑元件阵列所述恢复机器状态的信息；

电压调节器，其经连接以从所述输入电源接收电力并提供电力到输出电源轨，所述输出电源轨被配置为提供电力到所述处理设备，其中所述电压调节器被配置为与所述电源管理控制器通信，并且响应于确定所述电力质量低于阈值而将所述输出电源轨与所述输入电源断开；

电荷存储元件，其被配置为在所述输出电源轨与所述输入电源断开后，提供足够的临时电力到所述处理设备，以为所述处理设备供电，从而在所述多个非易失性逻辑元件阵列中存储所述机器状态。

2. 如权利要求 1 所述的计算设备装置，其中所述电源管理控制器被配置为响应于确定所述电力质量低于阈值，发送信号以实现用于所述处理设备的时钟的停止。

3. 如权利要求 1 所述的计算设备装置，其中所述电压调节器被配置为响应于将所述输出电源轨与所述输入电源断开，发送断开信号到所述电源管理控制器。

4. 如权利要求 3 所述的计算设备装置，其中所述电源管理控制器被配置为响应于接收到所述断开信号，发送备份信号到所述至少一个非易失性逻辑控制器。

5. 如权利要求 1 所述的计算设备装置，其中所述电压调节器被配置为检测所述电力质量升到高于所述阈值，并响应于此，发送良好电源信号到所述电源管理控制器。

6. 如权利要求 6 所述的计算设备装置，其中所述电源管理控制器被配置为响应于接收到所述良好电源信号，发送信号以提供电力到所述多个非易失性逻辑元件阵列和所述至少一个非易失性逻辑控制器，从而有助于所述机器状态的恢复。

7. 如权利要求 7 所述的计算设备装置，其中所述电源管理控制器被配置为确定加电完成，并且响应于此，发送信号以实现用于所述处理设备的时钟的释放，其中所述处理设备在确定所述电力质量低于所述阈值之前从所述机器状态恢复操作。

8. 如权利要求 1 所述的计算设备装置，其中所述电荷存储元件包括至少一个专用的管芯上的电容器。

9. 一种向处理设备供电的方法，所述处理设备具有多个非易失性逻辑元件阵列；多个易失性存储元件；以及至少一个非易失性逻辑控制器，其被配置为控制所述多个非易失性逻辑元件阵列，以存储由所述多个易失性存储元件表示的机器状态，并将存储的机器状态从所述多个非易失性逻辑元件阵列读出到所述多个易失性存储元件，该方法包括：

将电力从输入电源提供到输出电源轨，所述输出电源轨被配置为提供电力到所述处理设备；

电压或电流检测器感测来自所述输入电源的电力质量；

确定所述电力质量小于阈值；

响应于确定所述电力质量小于阈值：

将所述输入电源与所述输出电源轨断开，并且

使用来自电荷存储元件的电力，将存储在所述多个易失性存储元件中的数据备份到多个非易失性逻辑元件阵列中，所述电荷存储元件被配置为在所述输出电源轨与所述输入电源断开后，提供足够的临时电力到所述处理设备，以为所述处理设备供电，从而在所述多个非易失性逻辑元件阵列中存储所述机器状态。

10. 如权利要求 9 所述的方法，其进一步包括响应于确定所述电力质量低于所述阈值，停止用于所述处理设备的时钟。

11. 如权利要求 9 所述的方法，其进一步包括响应于将所述输出电源轨与所述输入电源断开，发送断开信号到电源管理控制器。

12. 如权利要求 11 所述的方法，其进一步包括响应于接收到所述断开信号，所述电源管理控制器发送备份信号到所述至少一个非易失性逻辑控制器。

13. 如权利要求 9 所述的方法，其进一步包括检测到所述电力质量升到高于所述阈值。

14. 如权利要求 13 所述的方法，其进一步包括电源管理控制器响应于检测到所述电力质量升到高于所述阈值，发送信号以提供电力到所述多个非易失性逻辑元件阵列和所述至少一个非易失性逻辑控制器，以有助于所述机器状态的恢复。

15. 如权利要求 14 所述的方法，其进一步包括：

所述电源管理控制器检测到加电完成，并且响应于此，发送信号以实现用于所述处理设备的时钟的释放；以及

所述处理设备在确定所述电力质量低于所述阈值之前从所述机器状态恢复操作。

16. 一种向处理设备供电的方法，所述处理设备具有多个非易失性逻辑元件阵列；多个易失性存储元件；以及至少一个非易失性逻辑控制器，其被配置为控制所述多个非易失性逻辑元件阵列，以存储由所述多个易失性存储元件表示的机器状态，并将存储的机器状态从所述多个非易失性逻辑元件阵列读出到所述多个易失性存储元件，该方法包括：

将电力从输入电源提供到输出电源轨，所述输出电源轨被配置为提供电力到所述处理设备；

电压或电流检测器感测来自所述输入电源的电力质量；

确定所述电力质量小于阈值；

响应于确定所述电力质量小于阈值：

将所述输入电源与所述输出电源轨断开，

使用来自电荷存储元件的电力，将存储在所述多个易失性存储元件中的数据备份到多个非易失性逻辑元件阵列中，所述电荷存储元件被配置为在所述输出电源轨与所述输入电源断开后，提供足够的临时电力到所述处理设备，以为所述处理设备供电，从而在所述多个非易失性逻辑元件阵列中存储所述机器状态；

停止用于所述处理设备的时钟；

响应于将所述输出电源轨与所述输入电源断开，发送断开信号到电源管理控制器；以及

响应于接收到所述断开信号，发送备份信号到所述至少一个非易失性逻辑控制器；

检测到所述电力质量升到高于所述阈值；以及

响应于确定所述电力质量升到高于所述阈值：

响应于检测到所述电力质量升到高于所述阈值，所述电源管理控制器发送信号以提供电力到所述多个非易失性逻辑元件阵列和所述至少一个非易失性逻辑控制器，以有助于所述机器状态的恢复，

所述电源管理控制器检测到加电完成，并且响应于此，发送信号以实现用于所述处理设备的时钟的释放，以及

所述处理设备在确定所述电力质量低于阈值之前从所述机器状态恢复操作。

在不一致供电期间的基于非易失性逻辑阵列的计算

技术领域

[0001] 本发明总体涉及非易失性存储器单元及其在系统中的使用，特别涉及与逻辑阵列组合，以提供非易失性逻辑模块。

背景技术

[0002] 许多便携式电子设备依靠电池操作，例如手机、数码相机 / 摄像机、个人数字助理、膝上型电脑和视频游戏。在不活动周期期间，设备可能不执行处理操作，并且可能被置于断电模式或待机功率模式以保存电力。在低功耗待机功率模式下，提供给电子设备内一部分逻辑的电源可能被关闭。然而，在待机功率模式期间漏电流的存在代表了对设计便携式、电池操作的设备的挑战。设备内的数据保持电路（例如触发器和 / 或锁存器）可以被用来在设备进入待机功率模式之前存储状态信息以便以后使用。数据保持锁存器（其还可以被称为阴影锁存器或气球锁存器）通常由单独的“常开”电源供电。

[0003] 用于在不活动周期期间减少漏电流的已知技术利用多阈值 CMOS(MTCMOS) 技术来实现阴影锁存器。在这种方法中，阴影锁存器利用厚栅氧化物晶体管和 / 或高阈值电压 (V_t) 晶体管来减少待机功率模式中的漏电流。阴影锁存器在正常操作期间（例如在有效功率模式期间）通常与电路的其余部分脱离，以维持系统性能。为了在“主从”触发器拓扑中保持数据，第三锁存器（例如阴影锁存器）可以被添加到主锁存器和从锁存器，用于数据保持。在其他情况下，从锁存器可以被配置为在低功耗操作期间作为保持锁存器操作。然而，仍然需要一些功率来保持所保存的状态。例如，参见美国专利 7,639,056 “Ultra Low Area Overhead Retention Flip-Flop for Power-Down Applications”，其通过引用并入到此。

[0004] 片上系统 (SoC) 是已经存在了很长时间的概念；基本方法是将越来越多的功能集成到给定的设备。这种集成可以采取硬件或解决方案软件的形式。传统上通过增加的时钟速率和更先进的工艺节点来实现性能增益。许多 SoC 设计将微处理器核或多个核与各种外围设备和存储器电路配对。

[0005] 能量收集（也被称为功率收集或能量采集）是从外部源得到、捕获能量并且存储用于小型无线自治设备（例如用在可佩带电子器件和无线传感器网络中的那些设备）的过程。收集的能量可以从各种源得到，例如：太阳能、热能、风能、盐度梯度和动能等。然而，典型的能量收集器为低能量的电子器件提供非常少量的功率。能量收集器的能量源作为周围背景而存在，并且可供使用。例如，温度梯度存在于内燃机的操作中，而在城区，由于无线电和电视广播等，环境中存在大量的电磁能量。

附图说明

[0006] 图 1 是根据本发明各种实施例配置的一部分示例片上系统 (SoC) 的功能方框图；

[0007] 图 2 是图 1 的 SoC 中使用的一个触发器群的更详细的方框图；

[0008] 图 3 是示出由铁电电容器展示的极化磁滞的图；

[0009] 图 4-7 是示出根据本发明的各种实施例配置的示例铁电非易失性位单元的原理

图和时序图；

[0010] 图 8-9 是示出根据本发明的各种实施例配置的另一个示例铁电非易失性位单元的原理图和时序图；

[0011] 图 10 是示出图 1 的 SoC 中使用的示例 NVL 阵列的方框图；

[0012] 图 11A 和图 11B 是图 10 的 NVL 阵列中使用的输入 / 输出电路的更详细的原理图；

[0013] 图 12A 是示出根据本发明的各种实施例配置的在读取周期期间的示例偏移电压测试的时序图；

[0014] 图 12B 是示出根据本发明的各种实施例配置的在偏移电压的示例扫描期间生成的直方图；

[0015] 图 13 是示出图 10 的 NVL 阵列中奇偶校验生成的原理图；

[0016] 图 14 是示出根据本发明的各种实施例配置的 NVL 阵列内的示例电源域的方框图；

[0017] 图 15 是根据本发明的各种实施例配置的在 NVL 阵列中使用的示例电平转换器的原理图；

[0018] 图 16 是示出根据本发明的各种实施例配置的在铁电位单元内使用感测放大器的电平移位的示例操作的时序图；

[0019] 图 17 是根据本发明的各种实施例配置的示例功率检测布置的方框图；

[0020] 图 18 是示出根据本发明的各种实施例配置的控制为具有 NVL 阵列的处理设备供电的示例操作的流程图；以及

[0021] 图 19 是根据本发明的各种实施例配置的包括 NVL 阵列的另一个示例 SoC 的方框图。

具体实施方式

[0022] 虽然现有技术系统使用保持锁存器来保持低功耗操作期间逻辑模块中的触发器的状态，但是仍然需要一些功率来保持状态。相比之下，在完全去除电源时，非易失性元件可以保持逻辑模块中的触发器的状态。这类逻辑元件在此将被称为非易失性逻辑 (NVL)。在 SoC（片上系统）内用 NVL 实现的微控制单元 (MCU) 可以具有在不损失功能的情况下停止、断电和加电的能力。在完全去除电源后，不需要系统复位 / 重启来恢复操作。这种能力对新兴的能量收集应用（例如近场通信 (NFC)、射频识别 (RFID) 应用和嵌入式控制和监测系统）来说是理想的，例如，在这些应用中，复位 / 重启过程的时间和功率消耗（或称为成本）会消耗许多可用能量，从而为有用的计算、感测或控制功能留下很少能量或没有留下能量。虽然本说明书讨论了包含可编程 MCU 以便对 SOC 状态机进行排序的 SOC，但本领域技术人员可以明白，NVL 可以应用于硬编码到普通逻辑门或基于 ROM、PLA 或 PLD 的控制系统的状态机。

[0023] 在一种方法中，SoC 包括一个或多个非易失性逻辑块。例如，基于非易失性逻辑 (NVL) 的 SoC 可以在接收到电力中断后备份其工作状态（所有触发器），在睡眠模式下具有零泄漏，以及在加电后需要小于 400ns 来恢复系统状态。

[0024] 没有 NVL，芯片将必须在至少一个低功耗保持状态（即使在待机模式下，该状态也需要持续的电源）下保持所有触发器供电，或者必须在加电后浪费能量和时间重启。对于

能量收集应用，NVL 是有用的，因为没有保存触发器 (FF) 状态所需的恒定电源，并且即使当间歇性的电源可用时，引导代码也会独自消耗所有的收集的能量。对于具有有限的冷却和电池容量的手持设备来说，具有“瞬时接通 (instant-on)”能力的零泄漏 IC（集成电路）是理想的。

[0025] 铁电随机存取存储器 (FRAM) 是与 DRAM（动态随机存取存储器）具有相似行为的非易失性存储器技术。可以访问每个单独的位，但是不像 EEPROM（电可擦除可编程只读存储器）或闪存，FRAM 不需要特殊的序列来写数据，其也不需要电荷泵来实现所需的较高编程电压。每个铁电存储器单元包含一个或多个铁电电容器 (FeCap)。各铁电电容器可以用作此处所述的 NVL 电路中的非易失性元件。

[0026] 图 1 是示出一部分计算设备的功能方框图，在这种情况下，示例片上系统 (SoC) 100 提供基于非易失性逻辑的计算特征。虽然术语 SoC 在此被用来指包含一个或多个系统元件的集成电路，但是本公开的教导可以应用于如下各种类型的集成电路，这些集成电路包含功能逻辑模块，例如锁存器、集成的时钟门控单元以及提供非易失性状态保持的触发器电路元件 (FF)。在大阵列的受控环境外部嵌入非易失性存储元件对可靠性和制造性提出了挑战。通常针对任意 NV- 存储器技术所需的最大读取信号裕量和原位裕量可测试性而设计基于 NVL 位单元的 NVL 阵列。然而，依据面积开销，向各个 NVL FF 增加可测试性特征可能是禁止的。

[0027] 为了摊销测试特征成本并提高可制造性，参考图 1 和图 2 的示例，布置多个非易失性逻辑元件阵列或 NVL 阵列 110 和多个易失性存储元件 220。至少一个非易失性逻辑控制器 106 被配置为控制该多个 NVL 阵列 110 以存储由多个易失性存储元件 220 表示的机器状态，并从该多个 NVL 阵列 110 将存储的机器状态读出到多个易失性存储元件 220。例如，至少一个非易失性逻辑控制器 106 被配置为生成用于保存机器状态至多个 NVL 阵列 110 或从多个 NVL 阵列 110 获取机器状态的控制序列。多路复用器 (MUX)212 被连接以将易失性存储元件 220 中的各个元件可变地连接到 NVL 阵列 110 中的一个或多个对应的单独的阵列。

[0028] 在所示的示例中，计算设备装置被布置在单个芯片上，此处是使用基于 FeCap（铁电电容）的位单元的 256b 迷你阵列 110（此处将被称为 NVL 阵列）实现的 SoC 100，基于 FeCap 的位单元被分散在整个逻辑群 (cloud) 中以当电源被移除时保存各种触发器 120 的状态。FF 120 的每一群 102-104 包括关联的 NVL 阵列 110。这种分散使得 NVL 阵列 110 的各个阵列在物理上被靠近布置并且连接以从易失性存储元件 220 中的对应各个元件接收数据。中央 NVL 控制器 106 控制所有的阵列和它们与 FF 120 的通信。虽然此处示出三个 FF 群 102-104，但是 SoC 100 可以具有全部由 NVL 控制器 106 控制的额外的或更少的 FF 群。SOC 100 可以被划分成多于一个的 NVL 域，在其中有专用 NVL 控制器用于管理每个单独的 NVL 域中的 NVL 阵列 110 和 FF 120。现有的 NVL 阵列实施例使用 256 位的迷你阵列，但是阵列可以根据需要具有更多或更少数量的位。

[0029] 使用包含如下电路系统的经修改的保持触发器 120 来实现 SoC 100，所述电路系统被配置为使得数据能够从多个非易失性逻辑元件阵列的各个阵列回写至多个触发器电路的各个电路。有各种已知的方法来实现保持触发器。例如，数据输入可以由第一锁存器锁存。当第一锁存器在待机功率模式中不工作时，耦合到第一锁存器的第二锁存器可以接收用于保持的数据输入。第一锁存器从第一电源线接收电力，在待机功率模式期间第一

电源线被关断。第二锁存器从第二电源线接收电力，在待机模式期间第二电源线保持接通。控制器接收时钟输入和保持信号，并提供时钟输出到第一锁存器和第二锁存器。保持信号的变化指示转换到待机功率模式。控制器继续将时钟输出保持在预定的电压电平，而第二锁存器继续在待机功率模式中从第二电源线接收电力，由此保持该数据输入。在美国专利 7,639,056“Ultra Low Area Overhead Retention Flip-Flop for Power-Down Applications”中更详细地描述了这种保持锁存器。

[0030] 图 2 示出示例保持触发器架构，该保持触发器架构不要求时钟在保持期间保持在特定的状态。在这种“时钟无关 (clock free)”NVL 触发器设计中，时钟值在保持期间是“无关的或称为不关心的 (don’t care)”。

[0031] 在 SoC 100 中，经修改的保持 FF 120 包括简单的输入和控制修改，以允许例如当系统被转换到断电状态时，每个 FF 的状态被保存在 NVL 阵列 110 中关联的 FeCap 位单元中。当系统恢复时，则将保存的状态从 NVL 阵列 110 转移回到每个 FF 120。通过实现特定的功率配置，可以提高功率节约和数据完整性。在这样的一种方法中，各保持触发器电路包括由第一电源域（例如下述示例中的 VDDL）供电的主逻辑电路部分（主级或主锁存器）以及由第二电源域（例如下述示例中的 VDDR）供电的从级电路部分。在这种方法中，在从多个 NVL 阵列向多个易失性存储元件回写数据期间，第一电源域被配置为断电而第二电源域有效。多个非易失性逻辑元件被配置为由第三电源域（例如下述示例中的 VDDN）供电，第三电源域在计算设备装置的常规操作期间被配置为断电。

[0032] 借助这种配置，可以实现多个电源域，这些电源域以可以专门设计以适合给定实现的方式独立地加电或断电。因此，在另一个方面，计算装置包括第一电源域和第二电源域，第一电源域被配置为向计算设备装置的切换逻辑元件供电，第二电源域被配置向如下逻辑元件供电，这些逻辑元件被配置为控制信号，以便将数据存储到多个非易失性逻辑元件阵列或从多个非易失性逻辑元件阵列读取数据。当多个易失性存储元件包括保持触发器时，第二电源域被配置为向保持触发器中的各触发器的从级供电。第三电源域为多个非易失性逻辑元件阵列供电。除了电源域，NVL 阵列可以被定义为与特定功能有关的域。例如，多个非易失性逻辑元件阵列中的至少一个阵列中的第一组可以与计算设备装置的第一功能关联，并且多个非易失性逻辑元件阵列中的至少一个阵列中的第二组可以与计算设备装置的第二功能关联。多个非易失性逻辑元件阵列中的至少一个阵列中的第一组的操作独立于多个非易失性逻辑元件阵列中的至少一个阵列中的第二组的操作。经这样配置，对分离的 NVL 阵列域或组的控制和处理的灵活性允许对计算设备整体功能进行更多粒度地控制。

[0033] 这种更具体的控制也可以应用于电源域。在一个示例中，第一电源域被划分成第一部分和第二部分，第一部分被配置为向与第一功能关联的切换逻辑元件供电，第二部分被配置为向与第二功能关联的切换逻辑元件供电。第一电源域的第一部分和第二部分被单独地配置为独立于第一电源域的其它部分而被加电或断电。类似地，第三电源域可以被划分成第一部分和第二部分，第一部分被配置为向与第一功能关联的非易失性逻辑元件阵列供电，第二部分被配置为向与第二功能关联的非易失性逻辑元件阵列供电。与第一电源域一样，第三电源域的第一部分和第二部分被单独地配置为独立于第三电源域的其它部分而被加电或断电。

[0034] 经如此配置，如果个别的功能未用于给定设备，则与未使用的功能关联的触发器

和 NVL 阵列可以分别与其他触发器和 NVL 阵列分开地断电和操作。功率和操作管理的这种灵活性允许关于功率使用和功能调整计算设备功能。这可以在下面的示例设计中进一步示出，该示例设计具有 CPU、三个 SPI 接口、三个 UART 接口、三个 I2C 接口和仅一个逻辑电源域 (VDDL)。逻辑电源域区别于保持或 NVL 电源域（分别为 VDDR 和 VDDN），虽然这些教导也可以应用到那些电源域。虽然这个示例设备只有一个逻辑电源域，但该设备的给定应用可能仅使用三个 SPI 单元中的一个、三个 UART 中的一个和三个 I2C 外围设备中的一个。为了允许应用优化 NVL 应用唤醒和睡眠时间以及能量成本，VDDL 电源域可以被划分为 10 个单独的 NVL 域（一个 CPU、三个 SPI、三个 UART、三个 I2C，共计 10 个 NVL 域），每个域可以独立于其他域而被启用 / 禁用。因此，客户可以针对其具体应用启用 CPU、一个 SPI、一个 UART 和一个 I2C 的 NVL 能力，同时禁用其他域。此外，该划分还允许时间以及能量的灵活性，并且不同的 NVL 域可以在不同的时间点保存和恢复状态。

[0035] 为了进一步增加灵活性，NVL 域可以与电源域交叠。参照上面的示例，可以定义四个电源域 :CPU、SPI、UART 和 I2C 各自一个（每个外围设备电源域具有三个功能单元），同时在每个外围设备域内定义三个 NVL 域，以及定义一个用于 CPU（再次总共有 10 个 NVL 域）。在这种情况下，除了控制每个电源域内的 NVL 域外，各电源域开启或关闭，以便增加功率节省和唤醒 / 睡眠时序的灵活性。

[0036] 此外，第一电源域、第二电源域和第三电源域中的每个电源域被配置为独立于第一电源域、第二电源域和第三电源域中的其他电源域而断电或加电。例如，一体的电源门 (power gate) 可以被配置为经控制以断电第一电源域、第二电源域和第三电源域中的每个电源域。如下表 1 中所述，第三电源域被配置为在计算设备装置的常规操作期间断电，而第二电源域被配置为在从多个非易失性逻辑元件阵列向多个易失性存储元件回写数据期间断电。第四电源域可以被配置为向实时时钟和唤醒中断逻辑供电。

[0037] 可以参考示出的示例 SoC 100 进一步理解这些方法，其中 NVL 阵列 110 和控制器 106 在被称为 VDDN 的 NVL 电源域上操作，并在常规操作期间关断。所有的逻辑、诸如 ROM（只读存储器）和 SRAM（静态随机存取存储器）的存储器块 107 以及 FF 的主级在被称为 VDDL 的逻辑电源域上。FRAM（铁电随机存取存储器）阵列直接连接到专用全局供电轨 (VDDZ)，其维持在 FRAM 所需的较高固定电压（即 VDDL< = VDDZ，其中 VDDZ 是固定的电源，而 VDDL 可以是变化的，只要 VDDL 保持在比 VDDZ 低的电势）。注意到，103 中所示的 FRAM 阵列通常包含集成的电源开关，该电源开关允许 FRAM 阵列根据需要而被断电，但是容易看出，没有内部电源开关的 FRAM 阵列可以与 FRAM 阵列外部的电源开关结合使用。保持 FF 的从级在被称为 VDDR 域的保持电源域上，以使得在待机操作模式下能够常规保持。表 1 总结了正常操作、系统备份到 NVL 阵列、睡眠模式、系统从 NVL 阵列恢复以及返回到正常操作期间的电源域操作。表 1 还指定待机空闲模式期间使用的域，可以在系统软件的控制下启动待机空闲模式，以便使用保持触发器的易失性保持功能进入功率降低的状态。在 108 处指示的一组开关用于控制各种电源域。可能存在多个开关 108，这些开关可以分布在整个 SoC 100 上并由软件控制，所述软件由 SoC 100 上的处理器和 / 或由 SoC 100 内的硬件控制器（未示出）执行。除了此处示出的三个域外，还可能存在额外的域，如后面将描述的。

[0038]

SoC 模式	触发器	触发器源	VDDL	VDDR	VDDN
常规操作	na	na	ON	ON	OFF
系统备份到 NVL	电源坏	外部	ON	ON	ON
睡眠模式	备份完成	NVL 控制器	OFF	OFF	OFF
系统从 NVL 恢复	电源好	外部	OFF	ON	ON
常规操作	恢复完成	NVL 控制器	ON	ON	OFF
待机保持模式	空闲	系统软件	OFF	ON	OFF

[0039] 表 1- 系统功率模式

[0040] 状态信息可以被保存在大型的集中式 FRAM 阵列中，但将需要更多的时间来进入睡眠模式，较长唤醒时间，过度路由以及由不能并行访问系统 FF 所造成的功率消耗。

[0041] 图 2 是 SoC 100 中使用的一个 FF 群 102 的更详细的方框图。在此实施例中，每个 FF 群包括多达 248 个触发器，并且每个 NVL 阵列被组织成 8×32 位的阵列，但在此实施例中一位用于奇偶校验。然而，在其它实施例中，触发器的数量和 NVL 阵列的组织形式可以具有不同的配置，如 4×m，16×m 等，其中 m 被选作匹配 FF 群的尺寸。在一些实施例中，各种群中的所有 NVL 阵列可以是相同大小，而在其他方法中，在同一 SoC 中可能有不同大小的 NVL 阵列。

[0042] 模块 220 是每个保持 FF 120 的更详细的原理图。若干信号具有由后缀“B”（指“bar”或 /）指示的反相版本，例如 RET 和 RETB、CLK 和 CLKB 等。每个保持 FF 包括主锁存器 221 和从锁存器 222。从锁存器 222 由反相器 223 和反相器 224 形成。反相器 224 包括由保持信号 (RET、RETB) 控制的一组晶体管，该组晶体管用于在低功耗睡眠周期期间保持 FF 状态，在该低功耗睡眠周期期间，电源域 VDDR 仍然接通，而电源域 VDDL 关闭，如上面和表 1 中所述。

[0043] NVL 阵列 110 与其在群 102 中服务的 248 个 FF 逻辑连接。一般而言，为了使数据能够从 NVL 阵列传送到 FF，各 FF 包括被配置为使得能够从多个 NVL 阵列 110 中的各个阵列回写数据的电路系统。在示出的示例中，在每个 FF 的从锁存器 222 上提供两个额外的端口，如模块 220 中所示。数据输入端口（门 225）被配置为将数据 ND 从 NVL 阵列 110 中的一个插入到关联的易失性存储元件 220。数据输入端口被配置为通过响应于在数据输入使能端口上接收到来自至少一个非易失性逻辑控制器 106 的更新信号 NU 以触发数据输入端口，从而允许与存储数据相关的信号从 NVL 阵列中的一个传送到关联的触发器电路的从级，来插入数据 ND。反相器 223 被配置为响应于接收到反相的 NVL 更新信号 NUZ 而被禁用，以避免三态反相器 223 和 NVL 数据端口输入三态反相器 225 之间的电冲突。

[0044] 更具体地，在示出的示例中，反相器 - 反相器反馈对 (223 和 224) 本身形成锁存器。这些反相器构成用于保持数据状态的非常稳定的配置，并将对抗改变锁存器状态的任意企图，除非至少一个反相器被禁用，以防止当通过其中一个数据端口尝试用下一状态改写当前状态时的电冲突。与常规触发器包括一个数据端口相比，示出的 NVL FF 220 包括访

问从锁存器 222 的两个数据端口。一个端口通过时钟控制的 cmos 传输门将数据从主级 221 传送到从级 222。当使用此端口更新从级 222 时，驱动到由 CLK 控制的传输门的输出节点的反相器 224 被禁用以避免电冲突，而反相器 223 被启用以将下一状态传送到锁存器的另一侧，使得锁存器的两侧具有下一状态，准备用于在时钟变低（对于上升沿 FF）时保持数据。

[0045] 由于相同的原因，当 ND 数据端口由转换到有效高状态的 NU 激活时，反相器 223 被禁用，以避免 ND 端口上的电冲突。第二反相器 224 被启用，以将下一状态传送到锁存器的另一侧，使得锁存器的两侧具有下一状态，以当 NU 变低时被锁存。在这个示例中，NU 端口没有以任何方式影响由时钟控制的其他数据端口。在双端口 FF 上，两个端口同时有效是非法的控制条件，产生的端口冲突意味着产生的下一状态将是不确定的。为了避免端口冲突，如果从状态被更新同时处于功能模式，则系统保持时钟处于无效状态。在保持模式中，RET 信号和 FF 内的支持电路用于在处于保持模式时防止电冲突，而不管 CLK 状态如何（见主级中由 RETB 控制的反相器）。

[0046] 如所示，这些额外的元件被布置在关联的 FF 的从级 222 中。然而，额外的晶体管不在 FF 的关键路径上，而且在此特定实施中对正常 FF 性能和功率（模拟数据）仅具有 1.8％和 6.9％的影响。当来自 NVL 阵列的数据在 ND(NVL- 数据）端口上有效时，NU(NVL- 更新）控制输入被脉冲到高电平持续一个周期以写入到 FF。NVL 阵列的三十一位数据输出扇出到 8 个三十一位 FF 组的 ND 端口。

[0047] 为了保存触发器状态，多路复用器被配置为传送来自多个易失性存储元件 220 中的多个单独的易失性存储元件元件的状态，以便基本同时存储在多个 NVL 阵列 110 中的单独一个 NVL 阵列中。例如，多路复用器可以被配置为连接到 N 组易失性存储元件，每组具有多个易失性存储元件中的 M 个易失性存储元件，并且连接到多个 NVL 阵列中 N×M 大小的 NVL 阵列。在此配置中，多路复用器将 N 组中的一组连接到 N×M 大小的 NVL 阵列，以将来自 M 个易失性存储元件的数据一次存储到 N×M 大小的 NVL 阵列的一行中。在示出的示例中，248 个 FF 的 Q 输出通过 31b 宽的 8-1 多路复用器 212 连接到 NVL 阵列 110 的 31b 并行数据输入。为了最小化 FF 加载，可以基于 FF 群的布局将多路复用器分解为更小的多路复用器，并且被放置得靠近它们服务的 FF。再次，NVL 控制器同步对 NVL 阵列的写入以及 8-1 多路复用器 212 的选择信号 MUX_SEL<2:0>。

[0048] 当 FF 在保持模式下操作时，计算设备的时钟 CLK 是“无关的或称为不关心的”，使得对于易失性存储元件来说，时钟 CLK 相对于每当 NU 信号有效时更新从级状态是不相关的，由此非易失性逻辑控制器被配置为控制和执行从易失性存储元件中的各元件到非易失性存储元件中的各元件的数据存储。换句话说，在保持模式期间 NVL 数据恢复的过程中不需要时钟 CLK 控制，但是一旦恢复系统状态，刚好在保持模式和功能模式之间的过渡之前，时钟 CLK 应被控制在系统级。在另一种方法中，当系统处于功能模式时，NVL 状态可以被恢复到易失性存储元件。在 VDDL 电源有效的这种情况下，在自 NVL 阵列的数据恢复期间，对于易失性存储元件，时钟 CLK 被保持在无效状态，由此非易失性逻辑控制器被配置为控制和执行从非易失性存储元件中的各元件到易失性存储元件中的各元件的数据传输。例如，对于基于上升沿 FF 的逻辑，系统时钟 CLK 通常被保持在低电平，而对于基于下降沿 FF 的逻辑，系统时钟 CLK 通常被保持在高电平。

[0049] 一般来说，为了从常规操作移动到系统备份模式，第一步是将系统时钟（一个或

多个）停止在无效状态以冻结机器状态，以便在备份进行时不改变。时钟被保持在无效状态直到备份完成。备份完成后，所有的电源域被断电，并且时钟的状态按照定义在睡眠模式中变为“无关的或称为不关心的”。

[0050] 当从 NVL 阵列恢复状态时，FF 被置于保持状态（参见下面的表 2），在保持状态中，只要 RET 信号是有效的，时钟就继续是“无关的或称为不关心的”（借助增加到每个保持 FF 的特殊晶体管，时钟可以是“无关的或称为不关心的”，并且由 RET 信号控制）。当恢复 NVL 状态时，触发器仍然处于保持模式，所以时钟仍然是“无关的或称为不关心的”。一旦 NVL 状态被恢复，控制系统时钟状态的机器逻辑的状态也将被恢复到它们原来在状态备份时所处的状态，这也意味着对于这个示例，将系统时钟置于无效状态的所有控制（包括易失性存储元件或 FF）现在已被恢复，使得在 NVL 数据恢复完成后，系统时钟将仍然处于无效状态。现在 RET 信号可以被停用，并且系统将在时钟停用的情况下处于静止，直到 NVL 控制器向电源管理控制器发送恢复完成的信号，响应于此，电源管理控制器将再次使能时钟。

[0051] 为了在恢复期间恢复触发器状态，NVL 控制器 106 读取 NVL 阵列 110 中的一行 NVL，然后为适当的触发器组脉冲 NU 信号。在系统恢复期间，保持信号 RET 被保持在高电平，并且在电源域 VDDL 未被供电的情况下从 ND 对从锁存器进行写入；此时，系统时钟 CLK 的状态是“无关的或称为不关心的”。在 VDDL ＝ 0V 且 VDDR ＝ VDD 的情况下将 FF 置于保持状态，以便抑制与假数据切换有关的过多功耗，所述假数据切换发生在 NVL 阵列读取操作期间更新每组 31 个 FF 时。适当修改的非保持触发器可以用于基于 NVL 的 SoC 中，其代价是在 NVL 数据恢复操作期间功耗较高。

[0052] 一旦 VDDL 上升，系统时钟 CLK 就应该从低电平开始，其后用 FF 中的更新信息继续正常的同步操作。可以以串行方式或并行方式或其任何组合完成 NVL 阵列和其相应的 FF 之间的数据传输，以权衡峰值电流和备份 / 恢复时间。因为直接访问被提供给由至少一个非易失性逻辑控制器控制的 FF，该至少一个非易失性逻辑控制器与用于计算设备装置的中央处理单元分离，所以对于 NVL 操作不需要来自微控制器处理单元 (CPU) 的干预；因此该实现是 SoC/CPU 架构不可知的。表 2 总结了 NVL 触发器的操作。

[0053]

模式	时钟(CLK)	保持(RET)	NVL 更新(NU)	保存的值
常规操作	脉冲	0	0	来自 D 输入
保持	X	1	0	Q 值
NVL 系统备份	0	0	0	来自 Q 输出
NVL 系统恢复	X	1	脉冲	NVL 单元位数据（ND）

[0054] 表 2-NVL 触发器真值表

[0055] 因为该至少一个非易失性逻辑控制器被配置为基于输入信号，以并行方式、顺序方式或其任何组合可变地控制向多个非易失性阵列的数据传输或从多个非易失性阵列的数据读取，所以关于针对特定需求调整系统操作规范，系统设计人员具有额外选择。例如，

因为在系统进入低功耗系统状态或从低功耗状态唤醒的时间期间，在 MCU SoC 上不会发生计算，所以最小化唤醒时间或进入睡眠的时间是有利的。另一方面，非易失性状态保持是功率密集型的，因为需要大量的能量来将状态保存到非易失性元件（例如铁电电容器），或从非易失性元件恢复状态。保存和恢复系统状态所需的功率能够超出电力输送系统的容量并引起问题，例如电迁移引起的电网退化、由于过度峰值电流汲取而导致的电池寿命减少、或者在电源系统上产生能够降低管芯上的信号完整性的高电平噪声。因此，允许系统设计者能够在这两个关注点之间进行平衡是期望的。

[0056] 在一个这种方法中，该至少一个非易失性逻辑控制器 106 被配置为通过用户接口 (U/I)125（例如，本领域技术人员已知的那些接口）接收输入信号。在另一个方法中，该至少一个非易失性逻辑控制器被配置为从可能正在执行应用的单独的计算元件 130 接收输入信号。在一个这种方法中，该单独的计算元件被配置为执行应用，以至少部分地基于用于计算设备装置 130 的功率和计算资源需求的确定来确定对多个非易失性阵列的读取序列。经如此配置，系统用户能够掌控系统状态存储和获取过程，以适应给定的设计。

[0057] 图 3 是示出由铁电电容器表现出的极化磁滞的绘图。铁电位单元的一般操作是已知的。当大多数材料被极化时，所感应的极化 P 几乎精确地与所施加的外部电场 E 成比例；所以该极化是线性函数，被称为介电极化。除了是非线性的，当所施加的电场 E 是零时，铁电材料还展示出如图 3 中所示的自发非零极化。铁电体的不同特征是：自发极化可以由所施加的电场反转；极化不仅依赖于当前的电场，而且还依赖于它的历史，从而产生磁滞回线。术语“铁电”用于指示具有自发磁化并且也表现出磁滞回线的铁磁材料的类似物。

[0058] 由于铁电材料的晶体结构中形成的半永久性电偶极子的影响，铁电电容器的介电常数通常远高于线性电介质的介电常数。当外部电场被施加到铁电电介质上时，偶极子趋向于将它们与电场方向对齐，在原子位置中产生小偏移，其导致晶体结构中电荷分布的偏移。去除电荷后，偶极子保持它们的极化状态。二进制“0”和“1”被存储在每个数据存储单元中，作为两个可能的电极化之一。例如在图中，可以使用负剩余极化 302 编码“1”，并可以使用正剩余极化 304 编码“0”，或反之亦然。

[0059] 已经以若干配置实现了铁电随机存取存储器。FeRAM 阵列中一晶体管、一电容器 (1T-1C) 的存储单元设计在结构上与广泛使用的 DRAM 中的存储单元的类似之处在于，两种单元类型包括一个电容器和一个存取晶体管。在 DRAM 单元电容器中使用线性电介质，而在 FeRAM 单元电容器中，介电结构包括铁电材料，典型地是锆钛酸铅 (PZT)。由于访问 DRAM 型阵列的开销，1T-1C 单元很少期望用于诸如 NVL 阵列 110 的小阵列中。

[0060] 四电容器、六晶体管 (4C-6T) 单元是更易于用在小阵列中的常见类型单元。现在将描述改进的四电容器单元。

[0061] 图 4 是示出包括四个电容器和十二个晶体管 (4C-12T) 的铁电非易失性位单元 400 的一个实施例的原理图。四个 FeCap 被布置成差分布置中的两对。FeCap C1 和 C2 串联连接以形成节点 Q 404，而 FeCap C1’ 和 C2’ 串联连接以形成节点 QB 405，其中经由位线 BL 将数据位写入到节点 Q 并存储在 FeCap C1 和 C2 中，并且经由反相位线 BLB 将数据位的反相写入到节点 QB 并存储在 FeCap C1’ 和 C2’ 中。感测放大器 410 耦合到节点 Q 和节点 QB，并被配置为当读取位单元时感测节点 Q、QB 上出现的电压差。感测放大器 410 中的四个晶体管被配置为两个交叉耦合的反相器，以形成锁存器。传输门 402 被配置为将节点 Q 耦合

到位线 B，而传输门 403 被配置为将节点 QB 耦合到位线 BLB。使用并联连接的 PMOS 器件和 NMOS 器件实现每个传输门 402、403。该布置降低了在写操作期间传输门两端的电压降，使得在写期间节点 Q、QB 上呈现较高的电压，由此将较高的极化赋予 FeCap。极板线 1(PL1) 耦合到 FeCapC1 和 C1'，而极板线 2(PL2) 耦合到 FeCap C2 和 C2'。极板线用于在读写操作期间向 FeCap 提供偏置。替代地，在另一实施例中，可以用 NMOS 传输门取代 cmos 传输门，所述 NMOS 传输门使用电压高于 VDDL 的传输门使能端。较高的电压幅度必须大于通常的 NMOS 的 Vt，以便将未退化的信号从位单元 Q/QB 节点传送到位线 BL/BLB 或从位线 BL/BLB 传送未退化的信号（即 V 传输 _ 门 _ 控制 (Vpass_gate_control) 必须大于 VDDL+Vt)。

[0062] 典型地，将存在位单元 400 的阵列。然后可能存在多列类似的位单元，以形成 n 行 ×m 列的阵列。例如在 SoC 100 中，NVL 阵列是 8×32；然而，如之前所讨论的，可以实现不同的配置。

[0063] 图 5 和图 6 是分别示出用于读取逻辑 0 的数据值和写入逻辑 0 的数据值的读取波形和写入波形的时序图。对 NVL 阵列的读取和写入是多周期过程，该多周期过程可以由 NVL 控制器控制并由 NVL 时钟同步。在另一实施例中，例如从触发信号开始，波形可以由固定或可编程的延迟排序。在常规操作期间，典型的 4C-6T 位单元容易受到由于在存储“1”的一侧上 FeCap 两端的恒定 DC 偏置导致的依赖时间的介电击穿 (TDDB) 的影响。在差分位单元中，由于还存储了数据值的反相版本，因此一侧或另一侧将始终存储“1”。

[0064] 为了避免 TDDB，当不访问该单元时，极板线 PL1、极板线 PL2、节点 Q 和节点 QB 被保持在静态的低值，如图 5、图 6 中在时间段 s0 期间所示。电源断开晶体管 MP 411 和 MN 412 允许感测放大器 410 响应于感测放大器使能信号 SAEN 和 SAENB 而在时间段 s0 期间与电源断开。钳位晶体管 MC 406 耦合到节点 Q 而钳位晶体管 MC' 407 耦合到节点 QB。钳位晶体管 406、407 被配置为在非访问时间段 s0 期间响应于清零信号 CLR 而将 Q 和 QB 节点钳位到约等于极板线上的低逻辑电压（在本实施例中等于 0 伏（地电位））的电压。以这种方式，在不访问位单元以进行读取或写入期间，没有电压被施加到 FeCap 两端，因此基本上消除了 TDDB。钳位晶体管还用来防止由于寄生漏电流而导致的在节点 Q 和 QB 上累计的任何杂散电荷 (stray charge)。杂散电荷的累计可能会使 Q 或 QB 上的电压升至 0 伏以上，从而在 Q 或 QB 和 PL1 和 PL2 之间的 FeCap 两端导致电压差。这会导致对 FeCap 剩余极化的非预期去极化，并可能潜在地破坏存储在 FeCap 中的逻辑值。

[0065] 在此实施例中，Vdd 是 1.5 伏，而地参考平面具有 0 伏的值。逻辑高具有大约 1.5 伏的值，而逻辑低具有大约 0 伏的值。使用与地为逻辑 0（低）以及 Vdd 为逻辑 1（高）不同的逻辑电平的其它实施例将节点 Q、QB 钳位到对应于静态极板线电压的电压，使得在不访问位单元时，FeCap 两端等效地没有电压。

[0066] 在另一实施例中，可以使用两个钳位晶体管。这两个晶体管中的每一个都用于将每个 FeCap 两端的电压钳位到不大于一个晶体管的 Vt（阈值电压）。每个晶体管用于短路 FeCap。在这种情况下，对于第一晶体管，一个端子连接到 Q 而另一个端子连接到 PL1，而对于晶体管二，一个端子连接到 Q 而另一个端子连接到 PL2。晶体管可以是 NMOS 或 PMOS，但是更有可能使用 NMOS。

[0067] 典型地，使用两晶体管解决方案的位单元不比一晶体管解决方案消耗明显更多的面积。单个晶体管解决方案假定 PL1 和 PL2 将保持在与到单个钳位晶体管的本地 VSS 连接

相同的地电位，这通常是一个很好的假定。然而，可能出现噪声或其他问题（尤其是在加电期间），噪声或其他问题可能导致 PL1 或 PL2 的毛刺或者在 PL1/PL2 驱动器输出和 VSS 之间具有 DC 偏移持续短暂的时期；因此，两晶体管设计可以提供更鲁棒的解决方案。

[0068] 为了读取位单元 400，将极板线 PL1 从低电平切换到高电平，同时保持极板线 PL2 为低电平，如时间段 s2 中所示。这在节点 Q、QB 上感应出电压，其值分别取决于 C1-C2 和 C1'-C2' 之间的电容器比率。感应的电压继而取决于在对位单元中的 FeCap 的上一次数据写入操作期间形成的每个 FeCap 的剩余极化。剩余极化实际上“改变”了每个 FeCap 的有效电容值，该有效电容值为 FeCap 如何提供非易失性存储。例如，当逻辑 0 被写入到位单元 400 时，C2 的剩余极化使其具有较低的有效电容值，而 C1 的剩余极化使其具有较高的有效电容值。因此，当通过切换极板线 PL1 到高电平同时保持极板线 PL2 为低电平而将电压施加到 C1-C2 两端时，节点 Q 上生成的电压符合公式 (1)。类似的公式也适用于节点 QB，但是 C1' 和 C2' 的剩余极化顺序是相反的，使得节点 Q 和 QB 上生成的电压提供对位单元 400 中所存储的数据值的差分表示，如图 5 中 502、503 处所示。

[0069] $$V(Q)=V(PL1)\left(\frac{C2}{C1+C2}\right) \qquad (1)$$

[0070] 然后在时间段 s3 期间使能本地感测放大器 410。在感测差分值 502、503 后，感测放大器 410 产生全轨信号 504、505。在时间段 s4 期间，通过断言传输门使能信号 PASS、PASSB 以使能传输门 402、403，将得到的全轨信号传送到位线 BL、BLB，并且由此响应于例如位于 NVL 阵列 110 的外围的锁存器使能信号 LAT_EN 而将该全轨信号传送到输出锁存器。

[0071] 图 6 是示出将逻辑 0 写入到位单元 400 的时序图。写入操作开始于在时间段 s1 期间将两个极板线提高到 Vdd。这被称为主存储方法。PL1 和 PL2 上的信号转换容性地耦合到节点 Q 和 QB，从而有效地将两个存储节点几乎一直拉到 VDD(1.5V)。在时间段 s2-s4 期间，在位线 BL、BLB 上提供数据并且由传输信号 PASS 使能传输门 402、403，以将数据位及其反相值从位线传输到节点 Q、QB。在时间段 s3、s4 期间，由感测放大器使能信号 SAEN、SAENB 使能感测放大器 410，以在写入数据驱动器在时间段 s2 期间在 Q/QB 上已强加足够的差分之后提供额外的驱动。然而，为了避免从感测放大器到 1.2v 驱动器电源的短路，在时间段 s3、s4 期间开启感测放大器之前，在时间段 s2 结束时关闭写入数据驱动器。在被称为二次存储方法的替换实施例中，在整个数据写入操作中，写入操作将 PL2 保持在 0v 或地。这可以在数据写入操作期间节省功率，但是将得到的读信号裕量降低了 50％，因为 C2 和 C2' 不再经由剩余极化保持数据，而只向 C1 和 C2FeCap 提供线性的电容负载。

[0072] 关键状态，例如在读取期间，在 s2 期间 PL1 高到 SAEN 高，在 s3 期间 SAEN 高脉冲，以及在写入期间，在 s3-4 期间 FeCap 的 DC 偏置状态，可以选择性地进行多个周期，以提供更高的鲁棒性而不减慢 NVL 时钟。

[0073] 对于基于 FeCap 的电路，从 FeCap 读取数据可能部分地去极化电容器。出于这个原因，从 FeCap 读取数据被认为本质上是破坏性的，即读取数据可能破坏 FeCap 的内容或至少降低数据完整性。出于这个原因，如果包含在 FeCap 中的数据在读取操作已经发生后被预计仍然有效，则数据必须写回到 FeCap。

[0074] 在某些应用中，可以指派特定的 NVL 阵列来存储在一段时间内不会改变的特定信息。例如，某些系统状态可以被保存为默认的返回状态，其中返回到该状态对完全重新启动

设备是优选的。用于最先进的超低功耗SoC的重新启动和配置过程会花费1000-10000个时钟周期或更长来达到控制被移交给主应用程序代码线程的点。对于电力是间歇性的、不可靠的并且数量有限的能量收集应用来说，这个启动时间变得至关重要。重新启动的时间和能量成本会消耗大部分或全部的可供计算使用的能量，从而阻止了可编程器件（例如MCU）被用在能量收集应用中。示例应用将是能量收集光开关。从按压光开关上的按钮收集的能量表示可供完成以下任务的整个能量：1）确定期望的功能（开/关或调光水平），2）将请求格式化为命令包，3）唤醒无线电并经由RF链路将包注射（squirt）到发光系统。由于严格的能量限制，具有硬编码状态机的已知定制ASIC芯片经常用于此应用，这使得系统改变缺乏灵活性且昂贵，因为每当期望任何改变时，必须设计和制造新的ASIC芯片。可编程MCU SOC将更适合该应用，除了启动过程的功率消耗消耗了大部分的可用能量，没有为执行所需的应用代码留下预算。

[0075] 为了解决这个问题，在一种方法中，多个非易失性逻辑元件阵列中的至少一个被配置为在完成给定量的启动过程后，存储表示计算设备装置状态的启动状态。这种方法中的至少一个非易失性逻辑控制器被配置为响应于检测到之前的系统复位或计算设备装置的功率丢失事件，而控制代表启动状态的数据从多个非易失性逻辑元件阵列中的至少一个到多个易失性存储元件中对应的易失性存储元件的恢复。为了在典型的读取/写入操作期间为NVL阵列节省功率，该至少一个非易失性逻辑控制器可以被配置为执行往返(round-trip)数据恢复操作，该操作在从单独的非易失性逻辑元件读取数据之后自动将数据回写到该单独的非易失性逻辑元件，而不需要完成分开的读取操作和写入操作。

[0076] 图7中示出了往返数据恢复操作的示例执行，其示出了位单元400上的回写操作，其中位单元被读取，然后被写入相同的值。如所示，通过将第一极板线PL1切换到高电平，以基于对应的铁电电容器位单元的铁电电容器的电容比率而在该对应的铁电电容器位单元的节点上感应出电压，在第一时间S1处开始启动从单独的非易失性逻辑元件读取数据。如果钳位开关用于将铁电电容器的节点接地，则在该第一时间S1处将清零信号CLR从高电平切换到低电平，以将该单独的非易失性逻辑元件的那些方面(aspect)从电气地松开。在第二时间S2处，感测放大器使能信号SAEN被切换到高电平，使得感测放大器能够检测节点上感应出的电压，并且提供对应于在该单独的非易失性逻辑元件中存储的数据的输出信号。在第三时间S3处，传输线PASS被切换到高电平，以打开传输门，从而提供对应于该单独的非易失性逻辑元件中存储的数据的输出信号。在第四时间S4处，第二极板线PL2被切换到高电平，从而在铁电电容器两端感应出极化信号，以将数据回写到对应于在该单独的非易失性逻辑元件中存储的数据的对应的铁电电容器位单元。为了该单独的非易失性逻辑元件到在其中存储有相同数据的非易失性存储状态，在第五时间S5处，第一极板线PL1和第二极板线PL2被切换到低电平，传输线PASS在第六时间S6处被切换到低电平，并且感测放大器使能信号SAEN在第七时间S7处被切换至低电平。如果钳位开关用于将铁电电容器的节点接地，则在第七时间处将清零信号CLR从低电平切换到高电平，以将该单独的非易失性逻辑元件的各方面钳位到电气地，从而帮助维持数据完整性，如在此所讨论的。与不同的且分开的读取和写入操作（读取，然后写入）所需的转换相比，这个过程包括总数更少的转换。这降低了总体的能量消耗。

[0077] 位单元400被设计为最大化Q/QB两端的读取差值，以便提供高度可靠的第一代的

NVL 产品。在每一侧上使用两个 FeCap，而不是使用一个 FeCap 和恒定的 BL 电容作为负载，因为这倍增了供给感测放大器的差分电压。感测放大器被置于位单元内，以防止由于节点 Q 和 BL 电容之间的电荷共享而导致的差值损失，并且避免传输门两端的电压降。感测的电压大约是 VDD/2，并且 HVT 传输门花费较长时间来将它们传递到 BL。位单元 400 帮助实现两倍的本领域已知的常规 FRAM 位单元的信号裕量，并且在 FeCap 两端不允许任何 DC 压力。

[0078] 图 5 和图 6 中所示的信号时序用于说明目的。各种实施例可以发送根据时钟速率、工艺参数、设备尺寸等而变化的信号序列。例如，在另一实施例中，控制信号的时序可以如下操作。在时间段 S1 期间 :PASS 从 0 变到 1 并且 PL1/PL2 从 0 变到 1。在时间段 S2 期间 :SAEN 从 0 变到 1，在此时间期间，感测放大器可以执行后面将描述的电平移位，或者为非电平移位的设计提供额外的驱动强度。在时间段 S3 期间 :PL1/PL2 从 1 变到 0，而波形的其余部分保持相同，但前移了一个时钟周期。此序列比图 6 示出的序列短一个时钟周期。

[0079] 在另一替换中，控制信号的时序可以如下操作。在时间段 S1 期间 :PASS 从 0 变到 1(BL/BLB、Q/QB 分别为 0V 和 VDDL)。在时间段 S2 期间 :SAEN 从 0 变到 1(BL/BLB、Q/QB 分别为 0V 和 VDDN)。在时间段 S3 期间 :PL1/PL2 从 0 变到 1(BL/Q 通过 PL1/PL2 耦合到地电平以上，并由 SA 和 BL 驱动器驱动回到低电平)。在时间段 S4 期间 :PL1/PL2 从 1 变到 0，而波形的其余部分保持相同。

[0080] 图 8-9 是示出铁电非易失性位单元 800(基于 2C-3T 自参考的 NVL 位单元) 的另一个实施例的原理图和时序图。与标准的 1C-1T FRAM 位单元相比，先前所述的基于 4-FeCap 的位单元 400 在感测放大器的每一侧上使用两个 FeCap，以得到具有双倍裕量的差分读取。然而，基于 4-FeCap 的位单元具有较大的面积，并且可能具有较大的变化，因为它使用了更多的 FeCap。

[0081] 通过使用自身作为参考 (此处称为自参考)，位单元 800 有助于在较小面积下实现类似差分 4-FeCap 的裕量。通过使用较少的 FeCap，它还比 4FeCap 的位单元具有较少的变化。典型地，单侧单元需要使用处于位单元的操作范围中间的参考电压。和双侧单元相比，这继而将读取裕量降低一半。然而，随着电路制造工艺的推进，参考值可能变得偏斜，从而进一步降低了读取裕量。自参考方案允许单侧单元与自身进行比较，由此提供更高的裕量。对此处所述的自参考单元的测试已经提供了相对于固定参考单元至少是两倍的裕量。

[0082] 位单元 800 具有两个 FeCap C1、C2，它们串联连接以形成节点 Q 804。极板线 1(PL1) 耦合到 FeCap C1 而极板线 2(PL2) 耦合到 FeCap C2。极板线用于在读写操作期间向 FeCap 提供偏置。传输门 802 被配置为将节点 Q 耦合到位线 B。使用并联连接的 PMOS 器件和 NMOS 器件实现传输门 802。这种布置降低了在写入操作期间传输门两端的电压降，使得在写入期间节点 Q、QB 上呈现较高的电压，由此将较高的极化赋予 FeCap。替换地，可以以提升的字线电压使用 NMOS 传输门。在这种情况下，PASS 信号将被提升一个 NFET 的 Vt(阈值电压)。然而，这可能会导致可靠性问题和过多的功耗。使用 CMOS 传输门向位单元增加额外的面积，但改善了速度和功耗。钳位晶体管 MC 806 耦合到节点 Q。钳位晶体管 806 被配置为在非访问时间段 s0 期间响应于清零信号 CLR 而将 Q 节点钳位到约等于极板线上的低逻辑电压 (在此实施例中为 0 伏 (地)) 的电压。以此方式，在没有访问位单元以便进行读取或写入期间，在 FeCap 两端没有施加电压，因此基本上消除了 TDDB 和非预期的部分去极化。

[0083] 节点 Q、极板线 PL1 和 PL2 的初始状态均为 0，如图 9 时间段 s0 处所示，所以当没有访问位单元时，在 FeCap 两端没有 DC 偏置。为了开始读取操作，PL1 被切换到高电平而 PL2 保持在低电平，如时间段 s1 期间所示。基于 FeCap 根据先前写入到单元的最后一个数据值所保持的极化，在节点 Q 上根据电容比率形成信号 902，如上面关于等式 1 所述的。通过将电压通过传输门 802 传输到位线 BL 上，然后响应于第二使能信号 EN1 而传送通过传输门 822，该电压被存储在位单元外部的读取电容器 820 上。注意到，在传输门 802、822 和 823 被使能之前，BL 和读取电容器被预充电到 VDD/2，以便当经由 BL 将 Q 上恢复的信号传输到读取存储电容器 820 和 821 时，最小化经由电荷共享而造成的信号损失。然后，在时间段 s2 期间，PL1 被切换回低电平，并且使用钳位晶体管 806 将节点 Q 放电。接下来，在时间段 s3 期间，PL2 被切换为高电平并保持 PL1 为低电平。新的电压 904 在节点 Q 上形成，但是这次是以相反的电容器比率形成。然后，经由传输门 823 将此电压存储在另一个外部读取电容器 821 上。因此，相同的两个 FeCap 用于读取高信号和低信号。然后，感测放大器 810 可以通过使用存储在外部读取电容器 820、821 上的电压来确定位单元的状态。

[0084] 典型地，将存在位单元 800 的阵列。图 8 中示出了经由位线 801 耦合到读取传输门 822、823 的一列位单元 800-800n。然后可以存在多列相似的位单元以形成 n 行 ×m 列的阵列。例如，在 SoC 100 中，NVL 阵列是 8×32；然而，如之前所述，可以实现不同的配置。例如读取电容和感测放大器可以位于存储器阵列的外围。

[0085] 图 10 是更详细地示出 NVL 阵列 110 的方框图。在大阵列的受控环境外嵌入非易失性元件呈现出可靠性和制造的挑战。如之前参照图 1 所述，在面积开销方面，向各 NVL FF 增加可测性特征可能是禁止的。为了摊销测试特征成本并提高可制造性，使用分散在整个逻辑群中的基于 FeCap 的位单元的 256b 迷你 -NVL 阵列 110 来实现 SoC 100，以便当电源被移除时保存各种触发器 120 的状态。FF 120 的每个群 102-104 包括关联的 NVL 阵列 110。中央 NVL 控制器 106 控制所有的阵列和他们与 FF 120 的通信。

[0086] 虽然可以以任意数量的 n 行 m 列配置实现 NVL 阵列，但是在这个示例中，用八行和三十二位列的位单元的阵列 1040 实现 NVL 阵列 110。每个单独的位单元（例如位单元 1041）耦合到由行驱动器 1042 提供的一组控制线。之前所述的控制信号，包括极板线 (PL1，PL2)、感测放大器使能信号 (SEAN)、传输门使能信号 (PASS) 和清零信号 (CLR)，均由行驱动器驱动。对于每一行位单元，有一组行驱动器。

[0087] 每个单独的位单元（例如位单元 1041）还经由位线耦合到一组输入 / 输出 (IO) 驱动器 1044。在此实现中，存在 32 组 IO 驱动器，例如 IO 驱动器组 1045。每个驱动器组产生输出信号 1047，当读取一行位线时，输出信号 1047 提供数据值。每条位线行进一列位单元的长度，并耦合到用于该列的 IO 驱动器。每个位单元可以被实现为例如 2C-3T 位单元 800。在这种情况下，单个位线将被用于每一列，并且感测放大器和读取电容器将位于 IO 驱动器模块 1044 中。在 NVL 阵列 110 的另一个实现中，每个位单元可以被实现为 4C-12T 位单元 400。在这种情况下，位线将是针对每列具有两个 IO 驱动器的差分对。比较器接收位线的差分对并产生最终的单个位线，这个位线被提供给输出锁存器。NVL 阵列 110 的其他实现可以结合将在下面更详细描述的行驱动器和 IO 驱动器使用其他已知的或以后开发的位单元。

[0088] 时序逻辑 1046 生成时序信号，时序信号用于控制读取驱动器，以针对每个读写操

作生成控制信号序列。可以使用同步或异步状态机，或者其它已知或以后开发的逻辑技术来实现时序逻辑 1046。一个潜在的替换实施例利用具有多个输出的延迟链，多个输出在期望的间隔处从延迟链“抽头 (tap)”，以生成控制信号。多路复用器可以用于为每个控制信号提供多个时序选择。另一个潜在的实施例使用可编程延迟发生器，可编程延迟发生器以期望的间隔使用连接到合适控制信号的专用输出产生边缘。

[0089] 图 11 是图 10 的 NVL 阵列中使用的一组输入 / 输出电路 1150 的更详细的原理图。返回来参照图 10，IO 模块 1044 中的三十二个驱动器的每个 IO 组 1045 类似于 IO 电路 1150。I/O 模块 1044 提供若干特征以有助于 NVL 位的可测试性。

[0090] 现在参照图 11，第一锁存器 (L1)1151 在读取期间用作输出锁存器，并且还与第二锁存器 (L2)1152 结合以形成扫描触发器。扫描输出 (SO) 信号被路由到写入驱动器模块 1158 中的多路复用器 1153，以在调试期间允许将扫描的数据写入到阵列中。扫描输出 (SO) 还耦合到下一组 IO 驱动器的扫描输入 (SI)，以形成 32 位扫描链，该 32 位扫描链可以用于读取或写入 NVL 阵列 110 的完整的一行比特。在 SoC 100 内，每个 NVL 阵列的扫描锁存器以串联方式连接以形成扫描链，从而允许使用扫描链访问所有的 NVL 阵列。替换地，可以以并行方式操作每个 NVL 阵列内的扫描链 (N 个阵列将生成 N 个链)，以减少每个链上的内部扫描触发器位的数量，以便加快扫描测试。链的数量和每条链上 NVL 阵列的数量可以根据需要而改变。典型地，SoC 100 内的所有的存储锁存器和触发器包括扫描链，以允许完成 SoC 100 的测试。扫描测试是公知的，此处不需要更详细地描述。在此实施例中，NVL 链与芯片上的逻辑链隔离，使得可以独立地训练链，并且 NVL 阵列可以在没有对逻辑链组织、实现或控制的任何依赖的情况下被测试。NVL 扫描链的最大总长度将总是小于逻辑链的总长度，因为 NVL 链长度被减小一个因子 (divisor)，该因子等于 NVL 阵列中的行数。在当前实施例中，每个 NVL 阵列有 8 个条目 (entry)，所以 NVL 扫描链的总长度是逻辑扫描链总长度的 1/8。这减少了访问和测试 NVL 阵列所需的时间，并且因此减少了测试成本。同样，它消除了确定逻辑触发器、它们在逻辑扫描链上的位置以及它们对应的 NVL 阵列位的位置 (识别阵列、行和列位置) 之间的映射的需要，从而极大简化了 NVL 测试、调试和失效分析。

[0091] 虽然扫描测试是有用的，但它并不为 SoC 100 的生产测试提供良好的机制，因为它可能花费相当大量的时间来扫描几百或几千个位，以便测试 SoC100 内的各种 NVL 阵列。这是因为不存在对 NVL 阵列内的位的直接访问。每个 NVL 位单元耦合到关联的触发器，并通过保存触发器的状态而被仅写入到关联的触发器。因此，为了将模式测试从关联的触发器加载到 NVL 阵列，必须使用扫描链设置对应的触发器。确定扫描链上的哪些位必须被设置或清除以便控制 NVL 阵列中的特定行的内容是复杂的任务，因为连接是基于硅管芯上的任意触发器组的物理位置而不是基于任何常规算法进行的。因此，触发器到 NVL 位置的映射无需被控制，并且通常有些随机。

[0092] 在 IO 驱动器 1150 内提供一种改进的测试技术。返回来参照图 1，NVL 控制器 106 具有状态机 (一个或多个)，以针对芯片上的所有 NVL 阵列执行快速通过 / 失败测试，从而筛选坏管芯。在这样一种方法中，至少一个非易失性逻辑控制器被配置为控制内建自测试模式，在该模式下，所有的 0 或所有的 1 被写入到多个 NVL 阵列中的至少一部分 NVL 阵列，然后确定从该至少一部分 NVL 阵列读取的数据是全 1 还是全 0。这通过以下步骤来完成：首先使用全 0/1 写入驱动器 1180 将全 0 或全 1 写入到一行，施加偏移扰动电压 (V_Off)，

然后使用并行读取测试逻辑 1170 读取同一行。如果来自数据锁存器 1151 的数据输出信号 (OUT) 为高电平，并且来自相邻列的 IO 驱动器的并行读测试逻辑“与 (AND)”门 G1 的信号 corr_1 为高电平，则来自“与”门 G1 的信号 corr_1 变为高电平。以这种方式，NVL 阵列 110 中的 32 组 I/O 模块 1150 的 G1 “与”门实现大的 32 输入“与”门，该 32 输入“与”门告诉 NVL 控制器，对于 NVL 阵列 110 的选定行，所有的输出是否为高。对于读取 0，“或 (OR)”门 G0 做相同的动作。以这种方式，NVL 控制器可以指导 SoC 100 内的所有的 NVL 阵列对选定行同时执行全 1 写入，然后指导所有的 NVL 阵列同时读取该选定行，并仅使用少量的控制信号提供通过 / 失败指示，而不将任何明确的测试数据从 NVL 控制器传输到 NVL 阵列。在典型的存储器阵列 BIST（内建自测试）实现中，BIST 控制器必须访问所有的存储器输出值，使得可以比较每个输出位和预期的值。假定在典型的硅 SOC 芯片上有几千个逻辑触发器，NVL 阵列输出的总数也可以以千计地度量。使用普通的 BIST 逻辑电路来测试这些阵列将是不切实际的，这是因为需要大量的数据连接和数据比较器。对于具有八行的 NVL 阵列来说，那么 NVL 测试方法会重复八次（重复的次数将根据阵列组织而变化。在一个示例中，10 条目 NVL 阵列实现将重复该测试方法 10 次），使得 SoC 100 中所有的 NVL 阵列可以被测试，以便在仅八个写入周期和八个读取周期中校正全 1 操作。类似地，SoC 100 中所有的 NVL 阵列可以被测试，以便在仅八个写入周期和八个读取周期中校正全 0 操作。所有的 NVL 阵列的结果可以被缩减为单个信号，该单个信号通过额外的“与”门和“或”门指示通过或失败，额外的“与”门和“或”门从每个 NVL 阵列接收 corr_0 和 corr_1 信号并产生单个信号 corr_0 和 corr_1 信号，或者 NVL 控制器可以查看每个单独的 corr_0 和 corr_1 信号。

[0093] 全 0/1 写入驱动器 1180 包括 PMOS 器件 M1、M3 和 NMOS 器件 M2、M4。器件 M1 和 M2 串联连接以形成耦合到位线 BL 的节点，而器件 M3 和 M4 串联连接以形成耦合到反相位线 BLB 的节点。由 NVL 控制器 106 生成控制信号“all_1_A”和反相的“all_1_B”。当在写入周期期间断言时，它们激活器件 M1 和 M4，使位线 BL 和 BLB 被上拉，以表示逻辑 1 的数据值。类似地，由 NVL 控制器 106 生成控制信号“all_0_A”和反相的“all_0_B”。当在写入周期期间有效时，它们激活器件 M2 和 M3，使位线 BL 和 BLB 被上拉，以表示逻辑 0 的数据值。以这种方式，32 个驱动器可操作为响应于控制信号而将全 1 写入到一行位单元，并响应于另一个控制信号而将全 0 写入到一行位单元。本领域技术人员可以很容易地设计其他电路拓扑来完成相同的任务。当前实施例是优选的，因为它仅需要 4 个晶体管来完成所需的数据写入。

[0094] 在正常写入操作期间，写入驱动器块 1158 接收要被存储到数据输入 (data_in) 信号上的数据位值。写入驱动器 1156、1157 将互补的数据信号耦合到位线 BL、BLB，由此耦合到选定的位单元。写入驱动器 1156、1157 由写入使能信号 STORE 使能。

[0095] 图 12A 是示出读取周期期间偏移电压测试的时序图。为了将扰动电压施加到位单元，在读取期间修改状态 s1。该附图示出用于读取数据值“0”（节点 Q) 的电压扰动测试；用于数据值“1”的电压扰动测试是相似的，但是将扰动电压注入到感测放大器的另一侧（节点 QB)。因此，在本实施例中，基于正在读取的逻辑值，扰动电压被注入到感测放大器的低压侧。传输门 1154、1155 耦合到位线 BL、BLB。由 NVL 控制器 106、由片外测试控制器或者经由外部生产测试器编程未示出（例如可能在片上，或者在片外，在外部测试器中）的数模转换器，以产生期望量的偏移电压 V_OFF。NVL 控制器 106 可以在 s1 时间段期间断言用于

位线侧存储“0”的 Vcon 控制信号，由此使能 Vcon 传输门 1154、1155，在 s1 期间使用 M2/M4 放电另一条位线，并在 s1 期间断言控制信号 PASS 以开启传输门 402、403。这将“0”存储侧的节点 Q/QB 上的电压初始化到偏移电压 V_Off，如 1202 处所示。该预充电的电压降低了在 s3 期间供给到 SA 的差值，如 1204 处所指示，从而使位单元更接近失效。对于快速生产测试，V_Off 可以被设置到所需的裕量值，然后使用 GO-1 的通过 / 失败测试可以用于筛选出任何失效的管芯。

[0096] 图 12B 示出了偏移电压扫描期间生成的直方图。可以通过扫描 V_Off 并使用读取周期序列扫描出读取数据位来研究位级失效裕量，如上所述。在这个示例中，最坏情况的读取裕量是 550mv，平均值是 597mv，而标准方差是 22mv。以这种方式，可以容易地确定 SoC 上每个 NVL 阵列中的所有位单元的操作特性。

[0097] 如上所述，在大阵列的受控环境外部嵌入非易失性元件呈现出可靠性和制造挑战。应当针对任意 NV- 存储器技术所需的最大读取信号裕量和原位可测试性来设计 NVL 位单元。然而，NVL 实现不能依靠像 SRAM 的内建自测试 (BIST)，因为 NVL 阵列分布在逻辑群内部。上述 NVL 实现包括由中央 NVL 控制器 106 控制的 NVL 阵列。当为满意的行为筛选管芯时，NVL 控制器 106 运行片上执行的一系列步骤而无需任何外部测试器介入。该测试器只需要发出开始信号，并施加对应于期望的信号裕量的模拟电压。控制器首先向 NVL 阵列中的所有位写入全 0 或全 1。然后，它开始读取阵列，一次一行。NVL 阵列读取操作不必紧跟 NVL 阵列写入操作。通常，高温烘烤周期被插入到数据写入操作和数据读取操作之间，以便加快依赖时间和温度的失效机制，使得可以在与制造相关的测试期间筛选出将影响长期数据保持性的缺陷。如上更详细描述的，该阵列包含对阵列的所有输出进行“与”运算和“或”运算的逻辑。这两个信号被发送到控制器。在读取每一行之后，控制器查看来自阵列的两个信号，并基于它先前所写的内容决定在扰动电压的存在下数据读取是否正确。如果数据不正确，则它发出失败信号到测试器，此时，测试器可以排除该管芯。如果该行通过，则控制器移动到阵列中的下一行。可以以正常 NVL 时钟频率并行测试所有阵列。这使得能够对 NVL 阵列进行高速片上测试，其中测试器只发出开始信号并提供期望的读取信号裕量电压，而 NVL 控制器在内建测试过程结束时报告通过，或者每当首次检测到失效行时生成失败信号。失败被立即报告，所以测试器可以在首次失败时中止测试过程，而不是浪费额外的测试时间来测试剩余的行。这是很重要的，因为用于所有非易失性存储器 (NVM) 的测试时间和测试成本通常占据了具有嵌入式 NVM 的 SOC 的整个测试成本的主要部分。如果 NVL 控制器激活“完成 (done)”信号而失败信号在测试过程期间的任意时刻未被激活，则经历测试的管芯已经通过了所需的测试。

[0098] 为了进一步的失效分析，控制器还可以具有调试模式。在这种模式下，测试器可以指定阵列和行数，然后 NVL 控制器可以仅对该行进行读取或写入。可以使用 NVL 扫描链扫描出读取内容。这种方法提供了对管芯上的任意 NVL 位的读取或写入访问，而无需 CPU 干预或无需使用长的复杂 SOC 扫描链，在该扫描链中，NVL 阵列位到各触发器的映射是随机的。此外，这可以与施加用于读取信号裕量确定的模拟电压配合完成，所以可以测量用于各个位的精确裕量。

[0099] 这些能力有助于使 NVL 实用，因为在没有可测试性特征的情况下，在产品中使用非易失性逻辑元件将是有风险的。此外，在管芯上进行通过 / 失败测试而与测试器交互最

小减小了测试时间，由此减小了成本。

[0100] 使用分布在逻辑群中的迷你阵列的NVL实现意味着类似ECC的复杂错误检测方法将需要在每个阵列基上使用大量的额外存储器列和控制逻辑，从面积的观点看，这可能是禁止的。然而，为了提供增强的可靠性水平，SoC 100的NVL阵列可以包括奇偶校验保护作为低成本错误检测方法，如现在将要更详细描述的。

[0101] 图13是示出NVL阵列110中奇偶校验生成的原理图，NVL阵列110示出了具有32列位(0:31)的示例NVL阵列，奇偶校验生成将输入数据值DATA_IN 1151与前一列的IO驱动器的类似“异或(XOR)”门的输出进行“异或”运算。再次参照图11A，NVL阵列的每个IO驱动器部分(例如部分1350)可以包含“异或”门1160。在行写入期间，列30中的“异或”门1160的输出是正在写入到位列0:30中的该行数据的整体奇偶校验值，并且被用于通过将其输出馈送给NVL迷你阵列的列31的数据输入端而将奇偶校验值写入到最后一列，如图11B中XOR_IN所示。

[0102] 以类似的方式，在读取期间，“异或”门1160将经由多路复用器1161来自读取锁存器1151的数据值DATA_OUT(参见图11)与前一列的IO驱动器的类似的“异或”门的输出进行异或运算。在位列30中的“异或”门1160的输出是从位列0:30读取的该行数据的整体奇偶校验值，并且用于与从位列31中读取的奇偶校验值在奇偶校验错误检测器1370中进行比较。如果从读取的数据中确定的整体奇偶校验值与从列31中读取的奇偶校验位不匹配，则声明奇偶校验错误。

[0103] 当检测到奇偶校验错误时，其指示存储的FF状态值是不可信的。因为NVL阵列通常是在SoC处于电源关闭状态后重新启动操作时进行读取，因此检测到奇偶校验错误指示需要执行完整的启动操作，以便重新生成正确的FF状态值。

[0104] 然而，例如，如果在关闭电源之前没有正确地存储FF状态，或者这是全新的器件，则可能存在不确定的状况。例如，如果NVL阵列为空，则通常所有位可能具有值0，或者它们可能全部具有值1。在全0的情况下，为全0生成的奇偶校验值将是零，这将匹配奇偶校验位值0。因此，奇偶校验测试将错误地指示FF状态是正确的且不需要启动操作，而其事实上需要启动操作。为了防止这种情形发生，例如奇偶校验位的反相版本可以由位线驱动器1365写入到列31。再次参照图11A，注意到，虽然用于列0-30的位线驱动器1156也将输入数据位反相，但是当data_in位被接收到时多路复用器1153将它们反相，所以结果是列0-30中的数据被存储为未反相的。在另一个实施例中，例如可以反相数据位而不反相奇偶校验错误。

[0105] 在全1的情况下，如果有偶数个列，则计算出的奇偶校验将等于0，并且反相值1将被存储在奇偶校验列中。因此，在偶数个数据列均为全1的NVL阵列中将不会检测到奇偶校验错误。为了防止这种情形发生，NVL阵列110被限制为具有奇数个数据列。例如在此实施例中，有31个数据列和1个奇偶校验列，共有32个位单元列。

[0106] 在一些实施例中，当NVL读取操作发生时，用于NVL阵列的控制逻辑使奇偶校验位被读取、反相和回写。这允许NVL阵列检测先前的NVL阵列写入何时是不完整的或无效的/受损的。剩余极化没有被单个读取周期彻底地擦除。通常，花费5-15个读取周期来完全去极化FeCap，或者将数据破坏得足以可靠地触发NVL读取奇偶校验。例如，如果由于电力丢失而在最后一个NVL存储操作期间仅写入8个NVL阵列行中的4行，则这将最有可能导

致不完整地捕捉先前的机器状态。然而，由于剩余极化，在最近的状态存储序列中未被写入的 4 行将可能仍然包含来自以前（例如两个 NVL 存储事件之前）的陈旧数据，而不是来自最近的 NVL 数据存储事件的数据。来自这 4 行的奇偶校验和陈旧数据将可能被读取作为有效数据而不是无效数据。当在下一个唤醒 / 加电事件期间从 NVL 阵列恢复机器状态时，这极有可能导致机器死锁或崩溃。因此，通过在读取每个条目之后回写反相的奇偶校验位，每一行的陈旧数据基本上被强制无效。

[0107] 将数据回写到 NVL 条目是功率密集型的，所以优选不要将数据回写到所有位，而只是奇偶校验位。阵列的当前实施例禁用 PL1、PL2 以及用于所有非奇偶校验位（即数据位）的感测放大器使能信号，以最小化该特征的寄生功耗。

[0108] 用这种方式，每次 SoC 从无电状态转换到加电状态时，可以作出有效确定，即从 NVL 阵列读取的数据包含有效的 FF 状态信息。如果检测到奇偶校验错误，则可以执行启动操作，而不是从 NVL 阵列恢复 FF 状态。

[0109] 返回来参照图 1，低功耗 SoC 100 具有多个电压和电源域，例如用于 NVL 阵列的 VDDN_FV、VDDN_CV，用于睡眠模式保持锁存器和阱供电的 VDDR，以及用于形成系统微控制器、各种外围设备、SRAM、ROM 等的逻辑块的体区的 VDDL，如之前关于表 1 和表 2 所述。FRAM 具有内部电源开关并连接到常开电源 VDDZ。此外，VDDN_FV 域可以被设计为在一个电压下操作，例如 FeCap 位单元所需的 1.5 伏，而 VDDL 和 VDDN_CV 域可以被设计为在较低的电压下操作以节约功率，例如 0.9-1.5 伏。这种实现需要在适当区域中使用电源开关 108、电平转换和隔离。现在将更详细地描述关于 NVL 模块 110 所需的隔离和电平转换这两方面。电路被设计为使得 VDDL/VDDN_CV 可以是小于或等于 VDDN_FV 的任意有效电压，并且该电路将正确运转。

[0110] 图 14 是示出 NVL 阵列 110 内电源域的方框图。可以如表 3 中所示布置各种逻辑和存储器模块。

[0111]

全芯片电压域	电压电平	
VDD	0.9-1.5	用于 VDDL、VDDR、VDDN_CV 电源开关的常开电源，以及常开逻辑（如果有的话）
VDDZ	1.5	用于 FRAM 并用于 VDDN_FV 电源开关的常开 1.5V 电源。FRAM 具有内部电源开关。
VDDL	0.9-1.5	所有逻辑，以及所有触发器的主级、SRAM、ROM、写入多路复用器、FF 输出上的缓冲器以及多路复用器输出：可变逻辑电压；例如 0.9 至 1.5V（VDDL）。该电源从 VDDL 电源开关的输出得到。
VDDN_CV	0.9-1.5	NVL 阵列控制和时序逻辑，以及 IO 电路，NVL 控制器。从 VDDN_CV 电源开关得到。
VDDN_FV	1.5	NVL 阵列字线驱动器电路 1042 和 NVL 位单元阵列 1040：与 FRAM 相同的电压。从 VDDN_FV 电源开关得到。
VDDR	0.9-1.5	这是数据保持域并包括保持触发器的从级、NVL 时钟上的缓冲器、触发器保持使能信号缓冲器以及 NVL 控制输出，例如触发器更新控制信号缓冲器和 NVL 数据输出上的缓冲器。从 VDDR 电源开关得到。

[0112] 表 3- 示例全芯片电源域

[0113] 使用单独的一组电源开关（例如之前所述的开关 108）控制表 3 中描述的电源域 VDDL、VDDN_CV、VDDN_FV 和 VDDR。然而，对于某些情况可能需要隔离。IO 缓冲器模块 1044 内的数据输出缓冲器在 NVL 逻辑电源域 VDDN_CV 中，因此在芯片正常操作期间，在域 VDDR（或 VDDL，这取决于具体的实现）为接通时，数据输出缓冲器可能保持关闭。在这样的情况下，ISO- 低隔离被实现为将所有这种信号接地。虽然 VDDN_CV 关闭，但是如果来自 VDDN_CV 域的任意信号是悬浮的（当 VDDN_CV 域断电时未被驱动）并且他们未被隔离，则连接到随机逻辑区域中的 VDDR（或 VDDL，这取决于具体的实现）域中的数据输出的逻辑可能在内部电路中的电源和地之间生成短路电流。这同样适用于 NVL 阵列的 correct_0/1 输出和扫描输出。此处总的想法是，当 NVL 阵列未被供电时，NVL 阵列的任意输出将被隔离。假使在芯片中存在常开逻辑，则必须在 VDD 域外围处使用输入隔离将从 VDDL 或 VDDN_CV 到 VDD 的所有信号隔离。额外的内建隔离存在于 NVL 触发器的 ND 输入处。这里，输入行进到传输门，传输门的控制信号 NU 由常开信号驱动。当预期输入是不确定的时，NU 被置为低电平，由此禁用 ND 输入端口。类似的内建隔离存在于 NVL 阵列的数据输入和扫描输入上。在 NVL 恢复期间当 VDDL 为关闭时将需要此隔离。此外，信号 NU 和 NVL 数据输入多路复用器使能信号

(mux_sel) 必须仅在 VDDR 域中被缓冲。这同样适用于保持使能信号。

[0114] 为了使能各种节电操作模式，VDDL 和 VDDN* 域在各种时间被关闭，并且隔离使得这是可能的而不产生短路电流。

[0115] 在 NVL 阵列的行进到 NVL 位单元的控制输入（例如：行使能信号、PL1、PL2、恢复信号、返回 (recall) 信号和清零信号）上需要从较低电压 VDDL 域到较高电压 VDDN 域的电平转换。通过允许能够在较低电压下操作的 SOC 逻辑模块和 NVL 逻辑门这样操作，这能够降低系统功耗。对于位单元阵列 1040 中的每一行位单元，有一组字线驱动器 1042 驱动每行位单元的信号，例如包括极板线 PL1、PL2、传输门使能信号 PASS、感测放大器使能信号 SAEN、清零使能信号 CLR 和电压裕量测试使能信号 VCON。位单元阵列 1040 和字线电路模块 1042 由 VDDN 供电。到 1042 的输入信号的电平移位由专用电平移位器处理（参见图 15），而到位单元阵列 1040 的输入的电平移位由 NVL 位单元内电路的特殊序列处理，而无需增加任何额外的专用电路到阵列数据路径或位单元。

[0116] 图 15 是 NVL 阵列 110 中使用的电平转换器 1500 的原理图。图 15 示出一个字线驱动器，其可能是一组字线驱动器 1402 的一部分。电平转换器 1500 包括 PMOS 晶体管 P1、P2 和 NMOS 晶体管 N1、N2，它们形成在 1.5 伏 VDDN 域中的区域 1502 中，用于字线驱动器 1042。然而，时序和控制模块 1046 中的控制逻辑位于 1.2v VDDL 域 (1.2v 用于表示范围可以从 0.9V 到 1.5V 的可变 VDDL 核心电源）中的区域 1503 中。1.2 伏信号 1506 表示由控制模块 1046 生成的行控制信号中的任一个，用于访问 NVL 位单元阵列 1040。反相器 1510 在区域 1503 中形成一对互补的控制信号 1511、1512，它们接着被路由到电平转换器 1500 中的晶体管 N1 和 N2。在操作中，当 1.2 伏信号 1506 变为高电平时，NMOS 器件 N1 将 PMOS 器件 P2 的栅极拉低，这使 P2 将信号 1504 上拉到 1.5 伏。类似地，当 1.2 伏信号 1506 变为低电平时，互补信号 1512 使 NMOS 器件 N2 将 PMOS 器件 P1 的栅极拉低，这上拉 PMOS 器件 P2 的栅极并允许信号 1504 变为低电平，大约 0 伏。NMOS 器件必须比 PMOS 强壮，所以转换器不被卡住。以这种方式，电平移位可以在电压域上完成，并且可以通过将控制逻辑（包括反相器 1510）放置在较低的电压域 1503 中，来节省功率。对于每个信号，控制器通过两个互补控制信号 1511、1512 耦合到每个电平转换器 1500。

[0117] 图 16 是示出使用铁电位单元内的感测放大器的电平移位操作的时序图。再次参照图 2，在写入操作期间，从多路复用器 212 提供给 NVL 阵列 110 的输入数据也需要从 1.2v 的 VDDL 域被电平移位到 FeCap 在 1.5 伏的 VDDN 域中最佳操作所需的 1.5 伏。这可以使用例如位单元 400 的感测放大器完成。再次参照图 4 和图 13，注意到，每个位线 BL（例如 BL 1352，其来自 1.2 伏的 VDDL 域）耦合到位单元 400 内的传输门 402 或 403。感测放大器 410 在 1.5v 的 VDDN 电源域中操作。现在参见图 16，注意到，在时间段 s2 期间，在位线 BL、BLB 上提供数据，并且在时间段 s2 期间由传输信号 PASS 使能传输门 402、403，以将数据位及其反相值从位线传输到差分节点 Q、QB。然而，如 1602 处所示，传输的电压电平仅限于小于 1.5 伏的电平，因为位线驱动器位于 1.2v 的 VDDL 域内。

[0118] 在写入数据驱动器（例如写驱动器 1156、1157）在时间段 s2 期间在 Q/QB 上强加足够的差分 1602 之后，在时间段 s3、s4 期间由感测放大器使能信号 SAEN、SAENB 使能感测放大器 410 以提供额外的驱动，如 1604 处所示。由于感测放大器由较高电压 (VDDN) 供电，因此感测放大器将对由写入数据驱动器在感测放大器两端建立的差分进行响应，并将感测

放大器的逻辑 0 侧钳位到 VSS(Q 或 QB)，而包含逻辑 1 的另一侧被上拉至 VDDN 电压电平。以这种方式，现有的 NVL 阵列硬件被重复使用，以提供 NVL 存储操作期间的电压电平移位功能。

[0119] 然而，为了避免从感测放大器到 1.2v 驱动器电源的短路，在时间段 s3、s4 期间开启感测放大器之前，在时间段 s2 期间结束时将写入数据驱动器与感测放大器隔离。这可以通过在时间段 s2 后去断言 STORE 信号而关闭位线驱动器来完成，和 / 或也可以通过在时间段 s2 后去断言 PASS 而禁用传输门来完成。

[0120] 使用上述布置，在处理或计算设备操作周期中的各个点处最大化功率节省或利用率的各种配置是可能的。在这样一种方法中，计算设备可以被配置为在一系列电源中断下连续操作而不丢失数据或重新启动。参照图 17 中所示的示例，如上所述的处理设备 1700 包括多个非易失性逻辑元件阵列 1710、多个易失性存储元件 1720 以及至少一个非易失性逻辑控制器 1730，该至少一个非易失性逻辑控制器 1730 被配置为控制所述多个非易失性逻辑元件阵列 1710 以存储由所述多个易失性存储元件 1720 表示的机器状态，并将存储的机器状态从所述多个非易失性逻辑元件阵列 1710 读取到所述多个易失性存储元件 1720。电压或电流检测器 1740 被配置为感测来自输入电源 1750 的电力质量。

[0121] 电源管理控制器 1760 与电压或电流检测器 1740 通信以接收关于来自电压或电流检测器 1740 的电力质量的信息。电源管理控制器 1760 还被配置为与至少一个非易失性逻辑控制器 1710 通信以提供实现将机器状态存储至多个非易失性逻辑元件阵列 1710 和从多个非易失性逻辑元件阵列 1710 恢复机器状态的信息。

[0122] 电压调节器 1770 经连接以从输入电源 1750 接收电力并提供电力到输出电源轨 1755，输出电源轨 1755 被配置为提供电力到处理设备 1700。电压调节器 1770 被进一步配置为与电源管理控制器 1760 通信，并且响应于确定电力质量低于阈值而将输出电源轨 1755 与输入电源 1750 断开（例如通过控制开关 1780）。

[0123] 电源管理控制器 1760 和电压或电流检测器 1740 与至少一个非易失性逻辑控制器 1730 和电压调节器 1770 一起工作，以独立于主计算路径而管理数据备份和恢复过程。在一个这样的示例中，电源管理控制器 1760 被配置为响应于确定电力质量低于阈值而发送信号，以实现用于处理设备 1700 的时钟的停止。接着，响应于将输出电源轨 1755 与输入电源 1750 断开，电压调节器 1770 可以发送断开信号到电源管理控制器 1760。响应于接收到断开信号，电源管理控制器 1760 发送备份信号到至少一个非易失性逻辑控制器 1710。在完成系统状态备份到 NVL 阵列后，可以从 SOC 去除电源，或者电源可以继续降级而没有机器状态的丢失的进一步问题。

[0124] 在不同的方法中，确定电力质量的各元件可以不同。例如，电压调节器 1770 可以被配置为检测电力质量升到高于阈值，并响应于此，发送良好电源信号到电源管理控制器 1760。作为响应，电源管理控制器 1760 被配置为发送信号以提供电力到多个非易失性逻辑元件阵列 1710 和至少一个非易失性逻辑控制器 1730，以有助于机器状态的恢复。电源管理控制器 1760 被配置为确定加电完成，并且响应于此，发送信号以实现用于处理设备 1700 的时钟的释放，其中处理设备 1700 在确定电力质量低于阈值之前从机器状态恢复操作。

[0125] 为了确保处理设备 1700 具有足够的功率来完成备份过程，电荷存储元件 1790 被配置为提供足够的临时电源到处理设备 1700，以在输出电源轨 1755 与输入电源 1750 断开

后为它供电足够长，从而在多个非易失性逻辑元件阵列 1710 中存储机器状态。电荷存储元件 1790 可以是被设计为存储这种应急电源的至少一个专用管芯上（或管芯外）的电容器。在另一种方法中，电荷存储元件 1790 可以是自然产生的寄生电荷在管芯中累积的电路，其中从电路到地的电荷耗散提供足够的功率来完成备份操作。

[0126] 图 18 是示出上述处理设备的操作的流程图。例如，示例处理设备具有多个非易失性逻辑元件阵列、多个易失性存储元件和至少一个非易失性逻辑控制器，该控制器被配置为控制多个非易失性逻辑元件阵列，以存储由多个易失性存储元件表示的机器状态，并将存储的机器状态从多个非易失性逻辑元件阵列读出到多个易失性存储元件。该方法包括将电力从输入电源提供到输出电源轨 1802，输出电源轨被配置为提供电力到处理设备。电压或电流检测器感测来自输入电源的电力质量 1804。该方法进一步包括确定电力质量小于阈值 1806，该确定能够通过电压或电流检测器、电源管理控制器或其他硬件或软件方面做出。响应于确定电力质量小于阈值，将输入电源从输出电源轨断开 1808，并且使用来自电荷存储元件的电力，将存储在多个易失性存储元件中的数据备份到多个非易失性逻辑元件阵列中 1810。电荷存储元件被配置为在输出电源轨与输入电源断开后，提供足够的临时电力到处理设备，以为处理设备供电，从而在多个非易失性逻辑元件阵列中存储机器状态。

[0127] 取决于配置，该方法能够以若干方式执行。在一种方法中，响应于将输出电源轨与输入电源断开，发送断开信号到电源管理控制器。在该方法中，响应于接收到断开信号，电源管理控制器发送备份信号到至少一个非易失性逻辑控制器。响应于确定电力质量低于阈值，停止处理设备的是时钟。

[0128] 该方法能够接着包括检测到电力质量升到高于阈值 1812。在一种方法中，响应于检测到电力质量升到高于阈值，电源管理控制器发送信号以提供电力到多个非易失性逻辑元件阵列和至少一个非易失性逻辑控制器，以有助于机器状态的恢复 1814。在从 NVL 阵列恢复数据后，电源管理控制器检测到加电完成 1816，并且响应于此，发送信号以实现用于处理设备的时钟的释放 1818。处理设备在确定电力质量低于阈值之前从机器状态恢复操作。

[0129] 经如此配置，当检测到电压或电流降落时，NVL 系统能够停止执行，在系统电压降落到低于最小操作电压之前，将系统状态备份到 NVL 存储装置中，并且在电力恢复后恢复系统状态。好像电力并未中断一样，该系统能够接着恢复计算。在电力丢失的间隔之间的无缝计算能够使能很多新的应用。

[0130] 系统示例

[0131] 图 19 是包括如上所述的 NVL 阵列的另一个 SoC 1900 的方框图。SoC 1900 的特征在于，Cortex-M0 处理器核 1902、通用异步接收器 / 传送器 (UART) 1904 和 SPI（串行外围接口）1906 接口以及 10KB ROM 1910、8KB SRAM 1912、64KB（铁电 RAM）FRAM 1914 存储器块，这些表征了商业超低功率 (ULP) 微控制器。基于 130nm FRAM 工艺的 SoC 使用单个 1.5V 电源、8MHz 的系统时钟和用于 NVL 操作的 125MHz 时钟。SoC 消耗 75uA/MHz 和 170uA/MHz，并且分别运行来自 SRAM 和 FRAM 的代码。备份和恢复 2537 个 FF 的整个系统状态的能量和时间成本分别只需要 4.72nJ 和 320ns 以及 1.34nJ 和 384ns，其为这类设备设置了行业标准。SoC 1900 为每个 NVL 位提供了测试能力，如上更详细地描述的，并且提供了 550mV 的原位读取信号裕量。

[0132] SoC 1900 具有由 10 个 NVL 阵列服务的 2537 个 FF 和锁存器。中央 NVL 控制器控

制所有阵列及其与 FF 的通信，如上面更详细地描述的。分布式 NVL 迷你阵列系统架构有助于摊销测试特征成本，从而实现仅 3.6％开销的 SoC 面积，同时具有每位 2.2pJ/0.66pJ 的异常低的系统级睡眠 / 唤醒能量成本。

[0133] 虽然本发明特别适用于例如在片上系统 (SoC) 上实现的微控制器 (MCU)，但是其也适用于其他形式的处理器。SoC 可以包含一个或多个模块，每个模块包括与由设计库提供的与预先设计的功能电路结合的定制设计的功能电路。

[0134] 虽然本发明已参照说明性实施例进行描述，但该描述并不旨在被解释为限制意义。在参考此说明后，本发明的其它各种实施例对本领域技术人员将是显而易见的。例如，其他便携式或移动系统（例如遥控器、门禁卡和钥匙扣、智能信用 / 借记卡和仿真器、智能手机、数字助理，以及现在已知的或以后开发的其他任何便携式或嵌入式系统）可以包括此处所述的 NVL 阵列，以允许从完全断电状态几乎立即恢复到完全操作状态。

[0135] 虽然在此描述了耦合到非易失性 FeCap 位单元的保持锁存器的实施例，但是在另一个实施例中，来自 NVL 阵列的非易失性 FeCap 位单元可以耦合到不包括低功耗保持锁存器的触发器或锁存器。在这种情况下，例如，系统将在全功率状态或另外的基于降低的电压或时钟速率的降低的功率状态以及完全断电状态之间转换。如上所述，在关闭电源之前，触发器和锁存器的状态将被保存在分布式 NVL 阵列中。当电源恢复时，将经由关联的 NVL 阵列位单元提供的输入初始化触发器。

[0136] 可以以硬件、软件、固件或其任意组合实现此公开中所述的技术。如果在软件中实现，则可以在一个或多个处理器（例如微处理器、专用集成电路 (ASIC)、现场可编程门阵列 (FPGA) 或数字信号处理器 (DSP)）中执行该软件。执行该技术的软件最初可以存储在计算机可读介质中并在处理器中加载和执行，计算机可读介质例如是光盘 (CD)、软盘、磁带、文件、存储器或其他任意的计算机可读存储设备。在一些情况下，该软件也可以在计算机程序产品中销售，其包括计算机可读介质和用于该计算机可读介质的包装材料。在一些情况下，可以通过可移动的计算机可读介质（例如软盘、光盘、快闪存储器、USB 密钥）、通过来自另一种数字系统上的计算机可读介质的传输路径等分布所述软件指令。

[0137] 虽然此处以顺序方式呈现和描述方法步骤，但是所示和所述的一个或多个步骤可以被省略、重复、同时执行和 / 或以不同于图中所示和 / 或此处所述的顺序的顺序执行。因此，发明的实施例不应被认为是限制图中所示和 / 或此处所述的具体步骤顺序。

[0138] 因此，可预期的是，权利要求将覆盖落入发明真实范围内的实施例的任意此类修改。

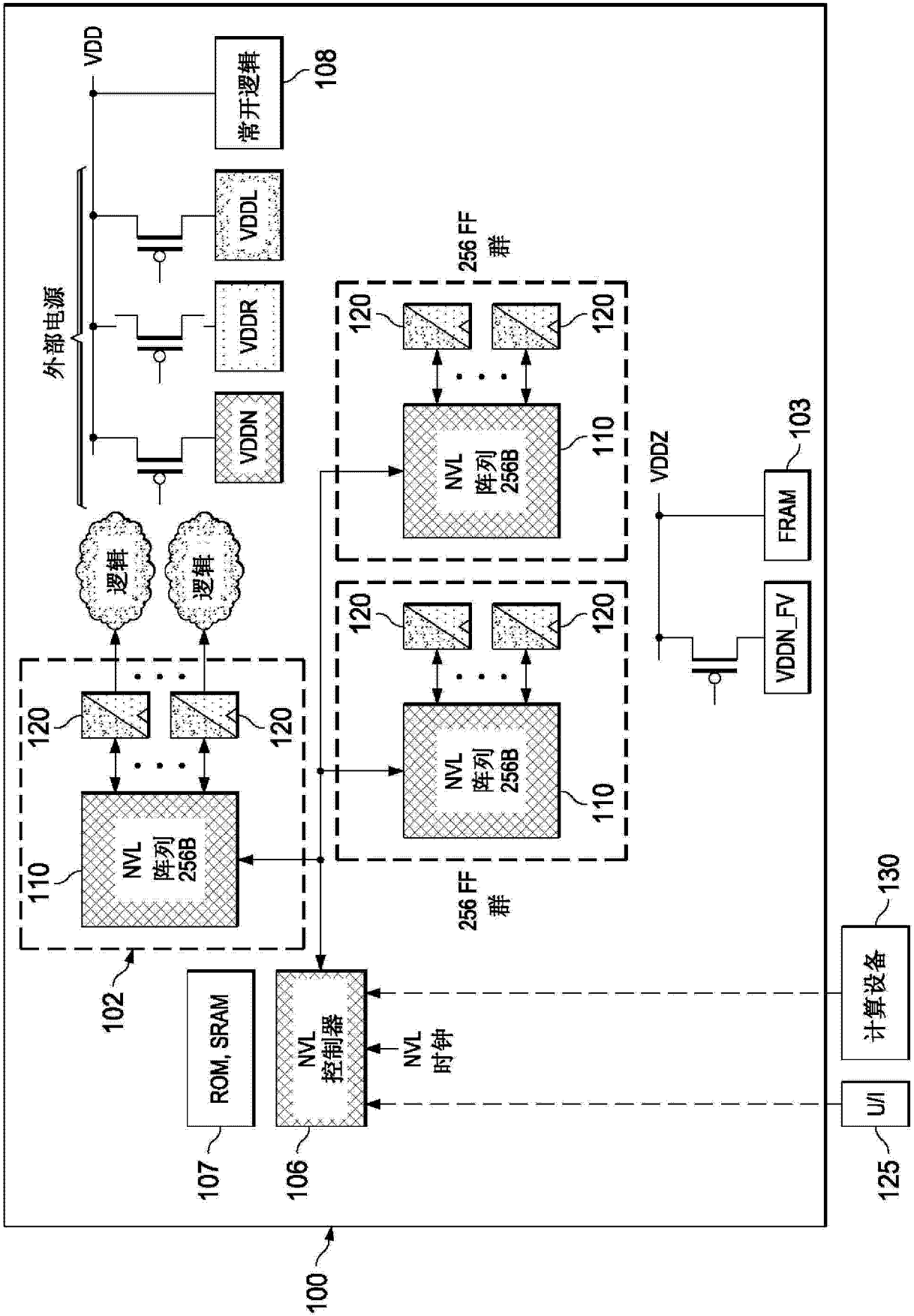

VDD
常开逻辑
108
外部电源
VDDL
VDDR
VDDN
逻辑
逻辑
120
120
NVL 阵列 256B
110
256 FF 群
120
120
NVL 阵列 256B
110
256 FF 群
VDDZ
FRAM
103
VDDN_FV
102
ROM, SRAM
107
NVL 控制器
106
NVL 时钟
计算设备
130
U/I
125
100

图 1

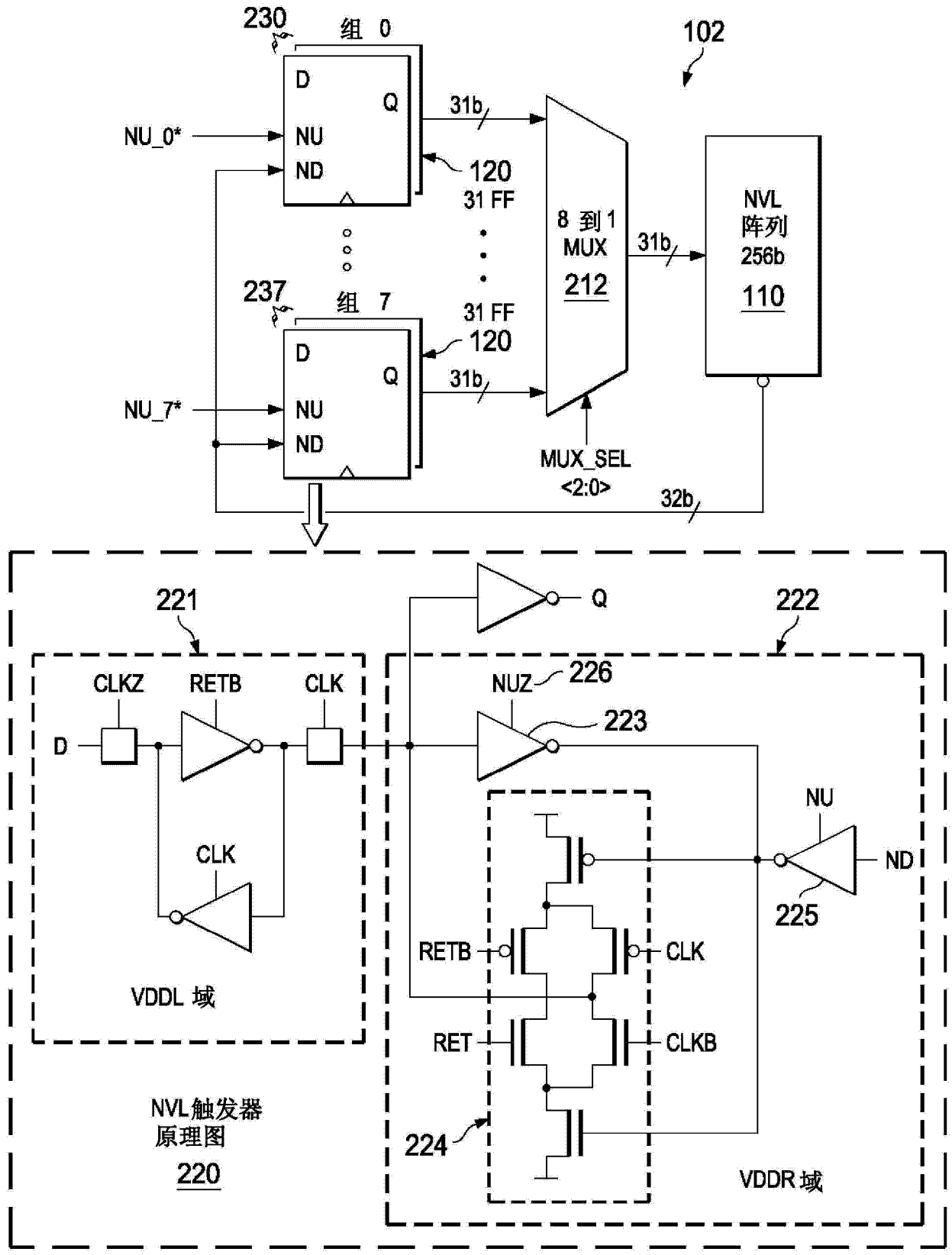

230
组 0
D
Q
NU_0*
NU
ND
31b
120
31 FF
237
组 7
NU_7*
8 到 1
MUX
212
102
NVL
阵列
256b
110
MUX_SEL
<2:0>
32b
221
222
CLKZ
RETB
CLK
D
Q
NUZ
226
223
NU
ND
225
RETB
CLK
RET
CLKB
224
VDDL 域
VDDR 域
NVL触发器
原理图
220

图 2

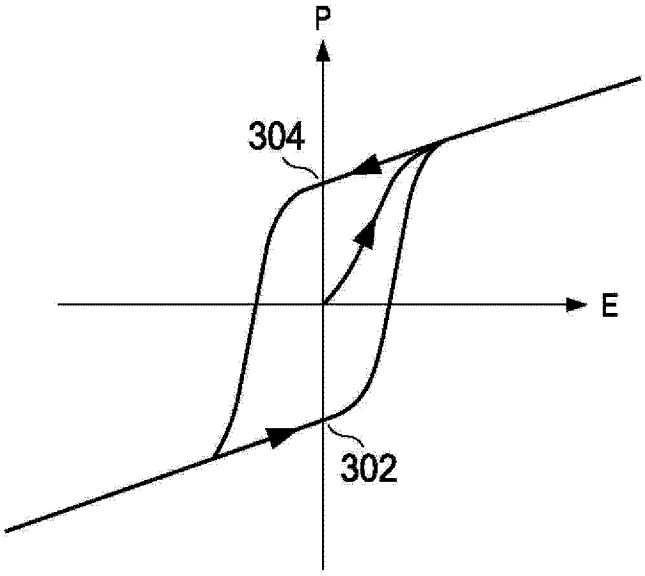


图 3

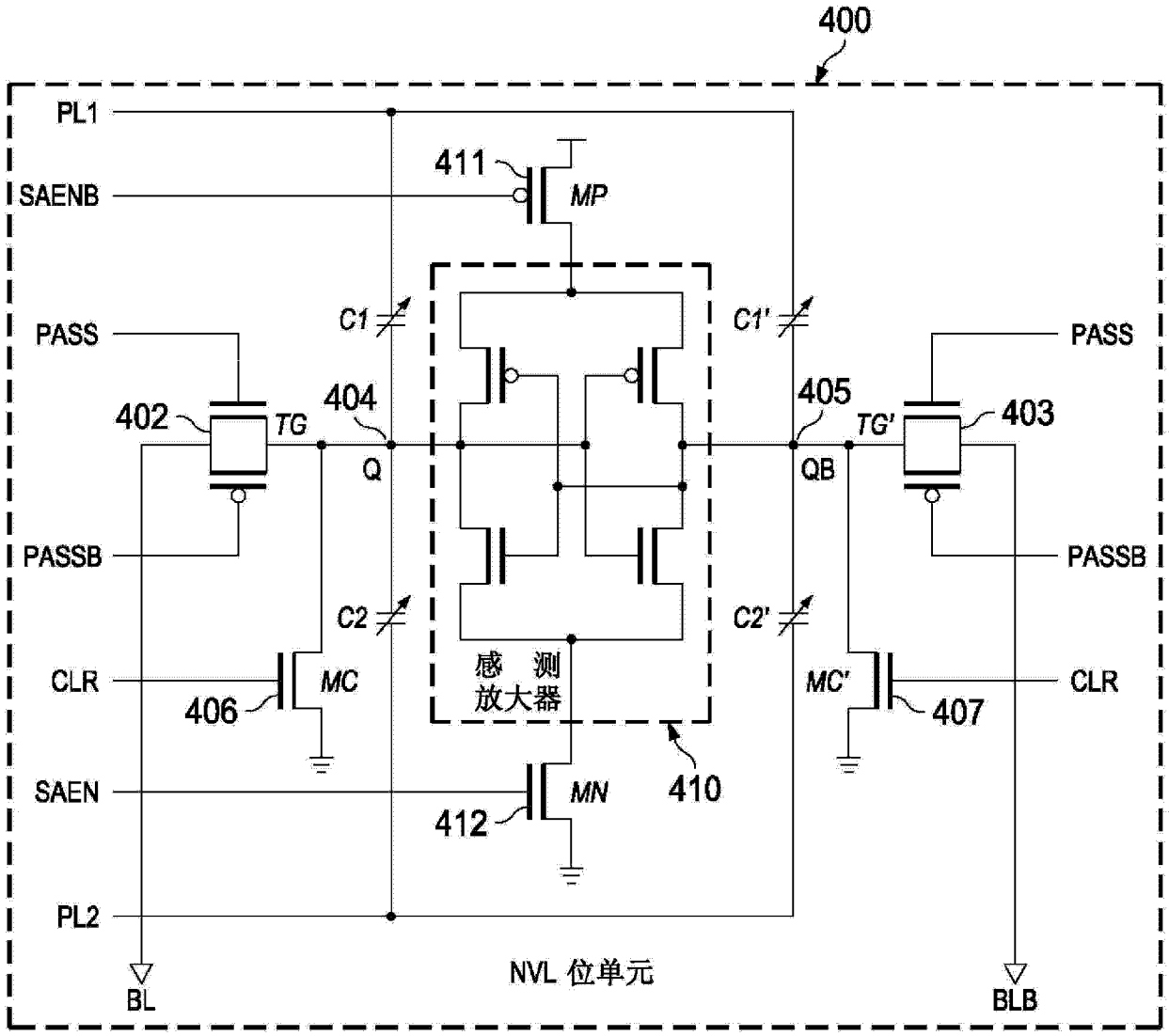

400
PL1
SAENB
411
MP
PASS
C1
C1'
PASS
402
TG
404
405
TG'
403
Q
QB
PASSB
PASSB
C2
C2'
CLR
MC
406
感 测
放大器
MC'
407
CLR
SAEN
MN
412
410
PL2
NVL 位单元
BL
BLB

图 4

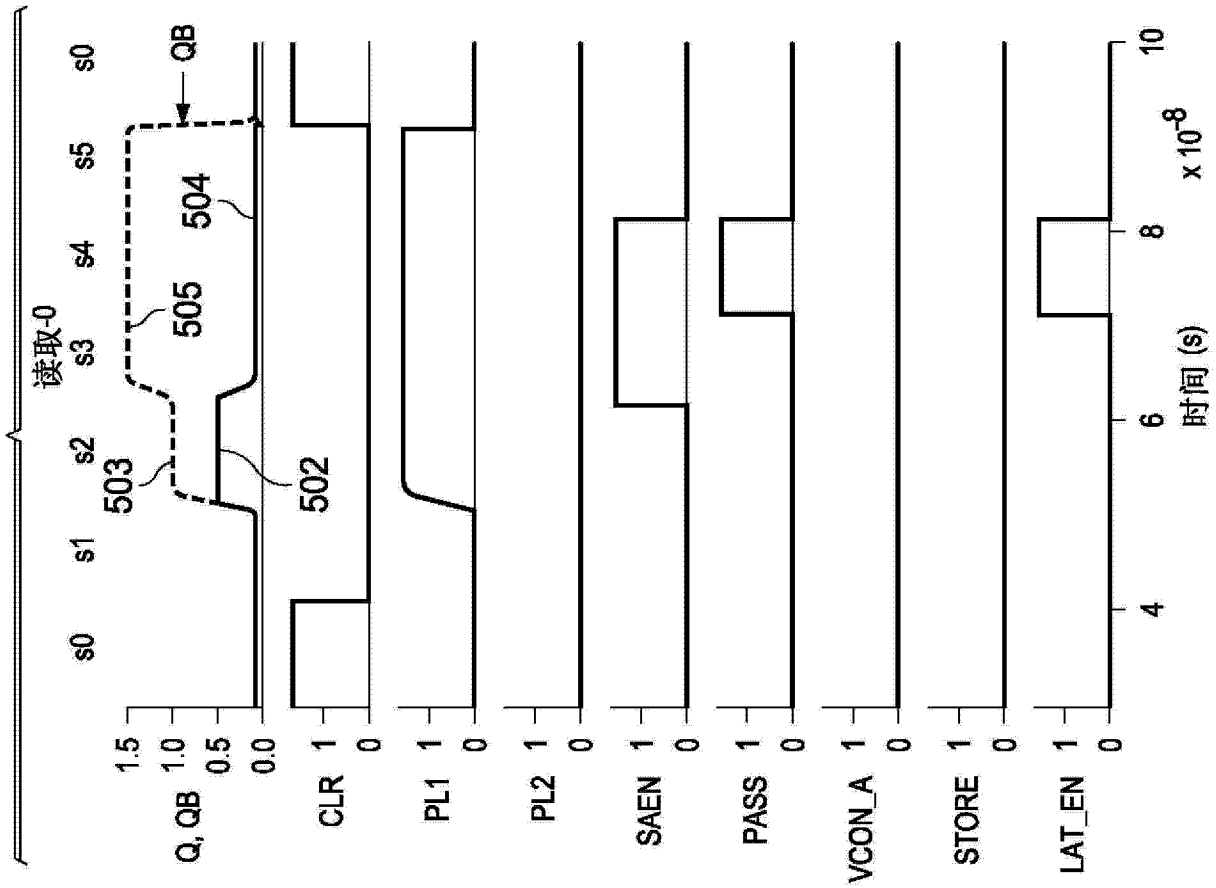

读取-0
s0
s1
s2
s3
s4
s5
s0
QB
503
502
505
504
Q, QB
1.5
1.0
0.5
0.0
CLR
PL1
PL2
SAEN
PASS
VCON_A
STORE
LAT_EN
1
0
4
6
8
10
x 10-8
时间 (s)

图 5

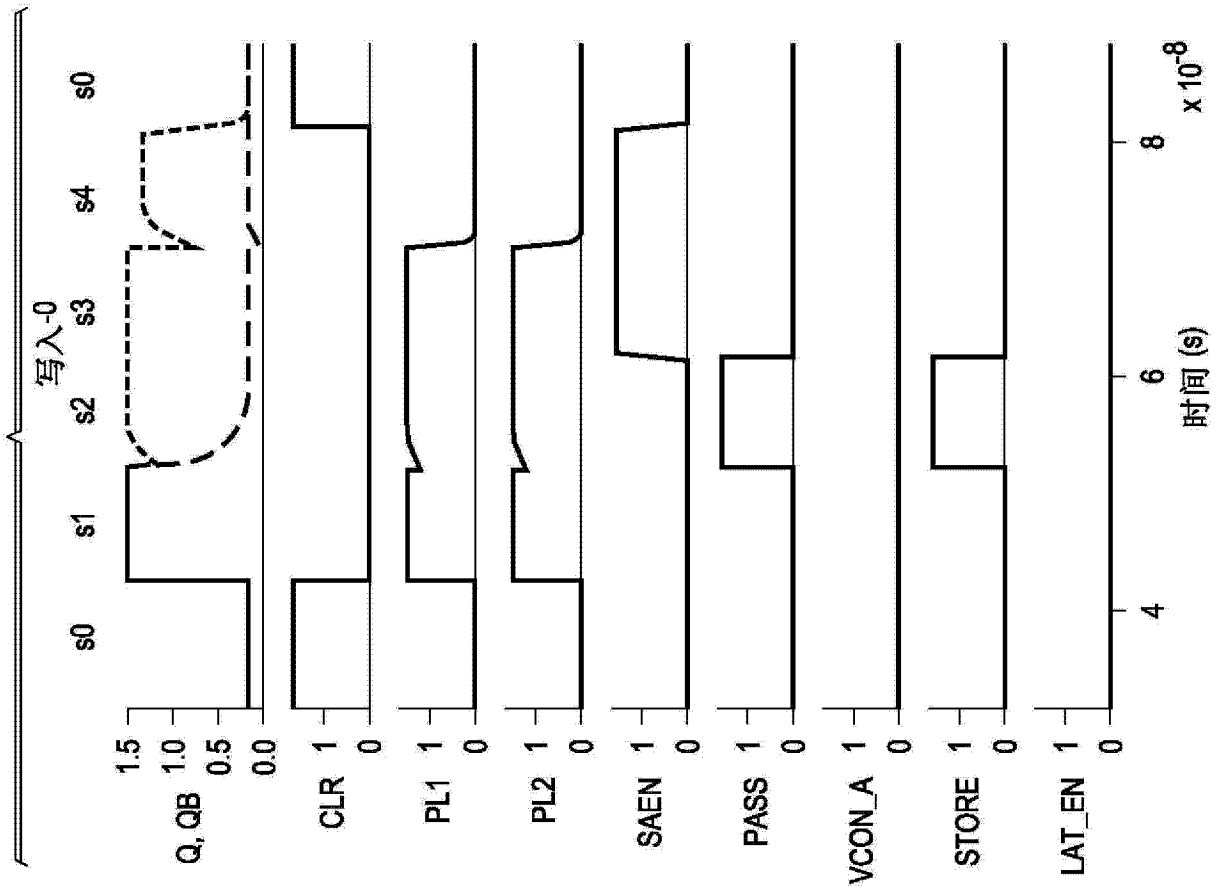

写入-0
s0
s1
s2
s3
s4
s0
Q, QB
1.5
1.0
0.5
0.0
CLR
PL1
PL2
SAEN
PASS
VCON_A
STORE
LAT_EN
1
0
4
6
8
时间 (s)
x 10^-8

图 6

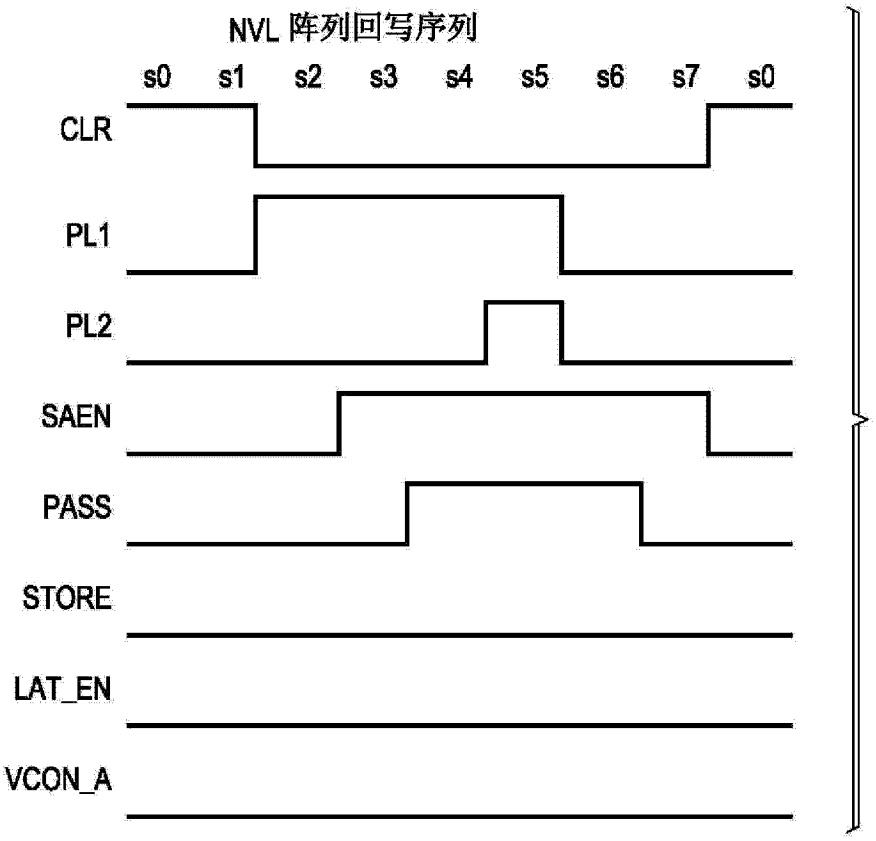

NVL 阵列回写序列
s0
s1
s2
s3
s4
s5
s6
s7
s0
CLR
PL1
PL2
SAEN
PASS
STORE
LAT_EN
VCON_A

图 7

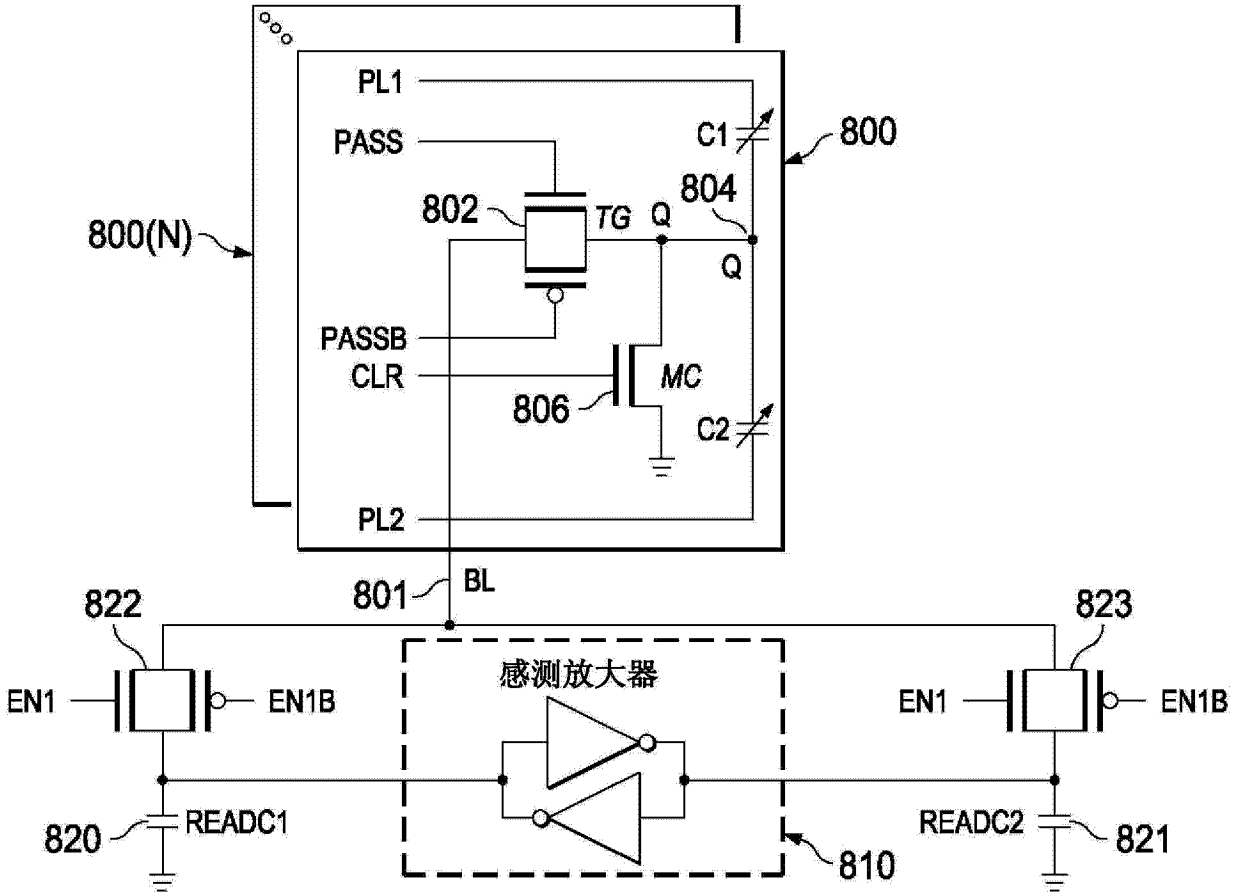

PL1
PASS
C1
800
802
TG
Q
804
800(N)
Q
PASSB
CLR
MC
806
C2
PL2
801
BL
822
823
感测放大器
EN1
EN1B
EN1
EN1B
820
READC1
READC2
821
810

图 8

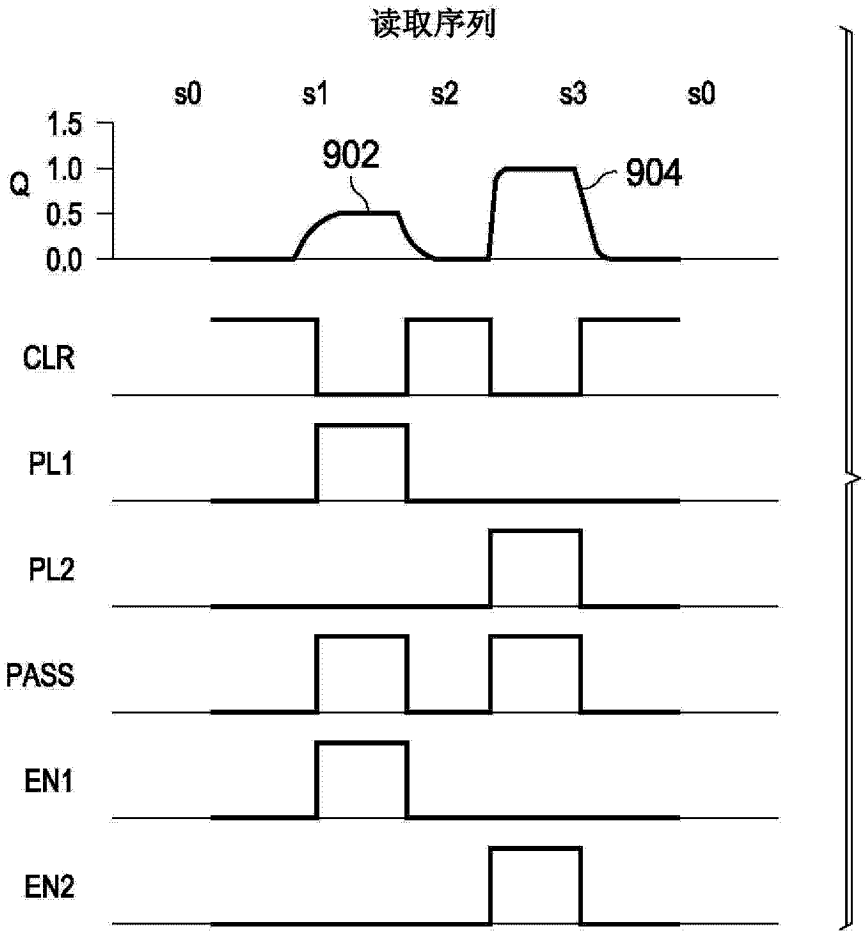


图 9

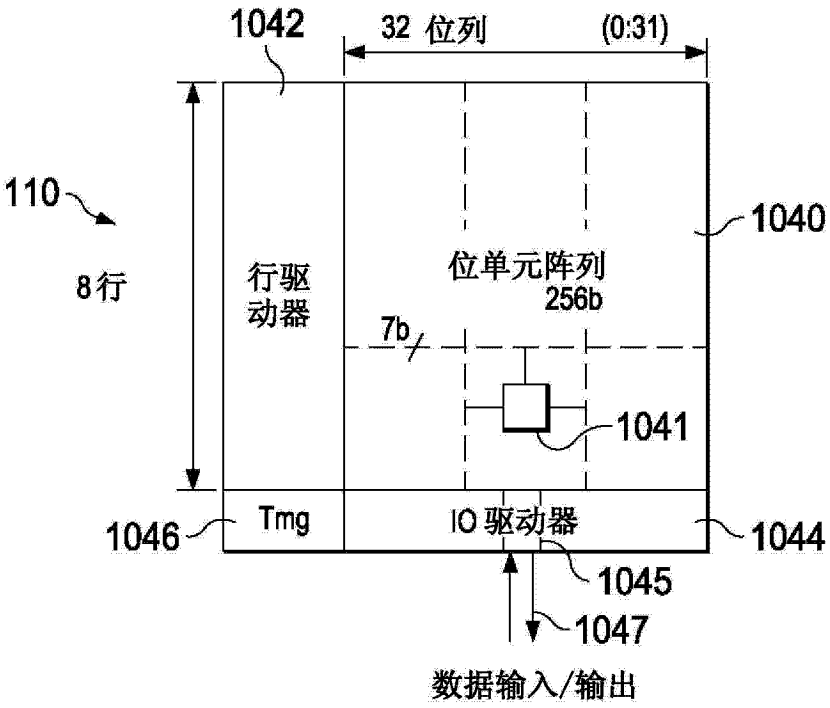

1042
32 位列
(0:31)
110
8行
行驱动器
位单元阵列
256b
1040
7b
1041
1046
Tmg
IO 驱动器
1044
1045
1047
数据输入/输出

图 10

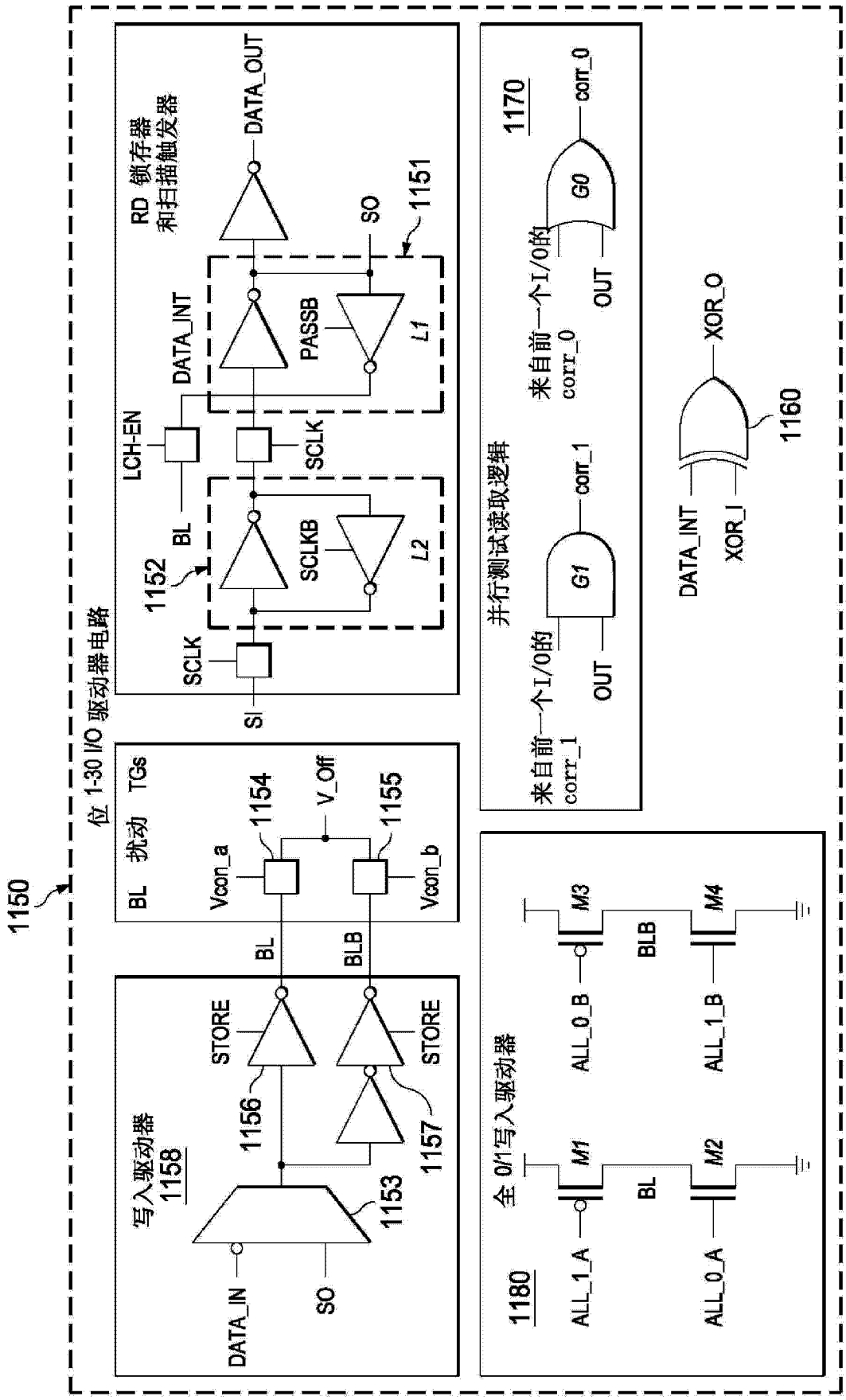

1150
位 1-30 I/O 驱动器电路
RD 锁存器
和扫描触发器
DATA_OUT
1151
SO
SI
LCH-EN
DATA_INT
PASSB
L1
SCLK
BL
1152
SCLKB
L2
BL 扰动 TGs
Vcon_a
1154
V_Off
1155
Vcon_b
BL
BLB
写入驱动器
1158
STORE
1156
1157
1153
DATA_IN
SO
并行测试读取逻辑
1170
来自前一个I/O的
corr_1
G1
corr_1
OUT
来自前一个I/O的
corr_0
G0
corr_0
OUT
DATA_INT
XOR_I
XOR_O
1160
全 0/1写入驱动器
1180
ALL_1_A
M1
ALL_0_B
M3
BL
BLB
ALL_0_A
M2
ALL_1_B
M4

图 11A

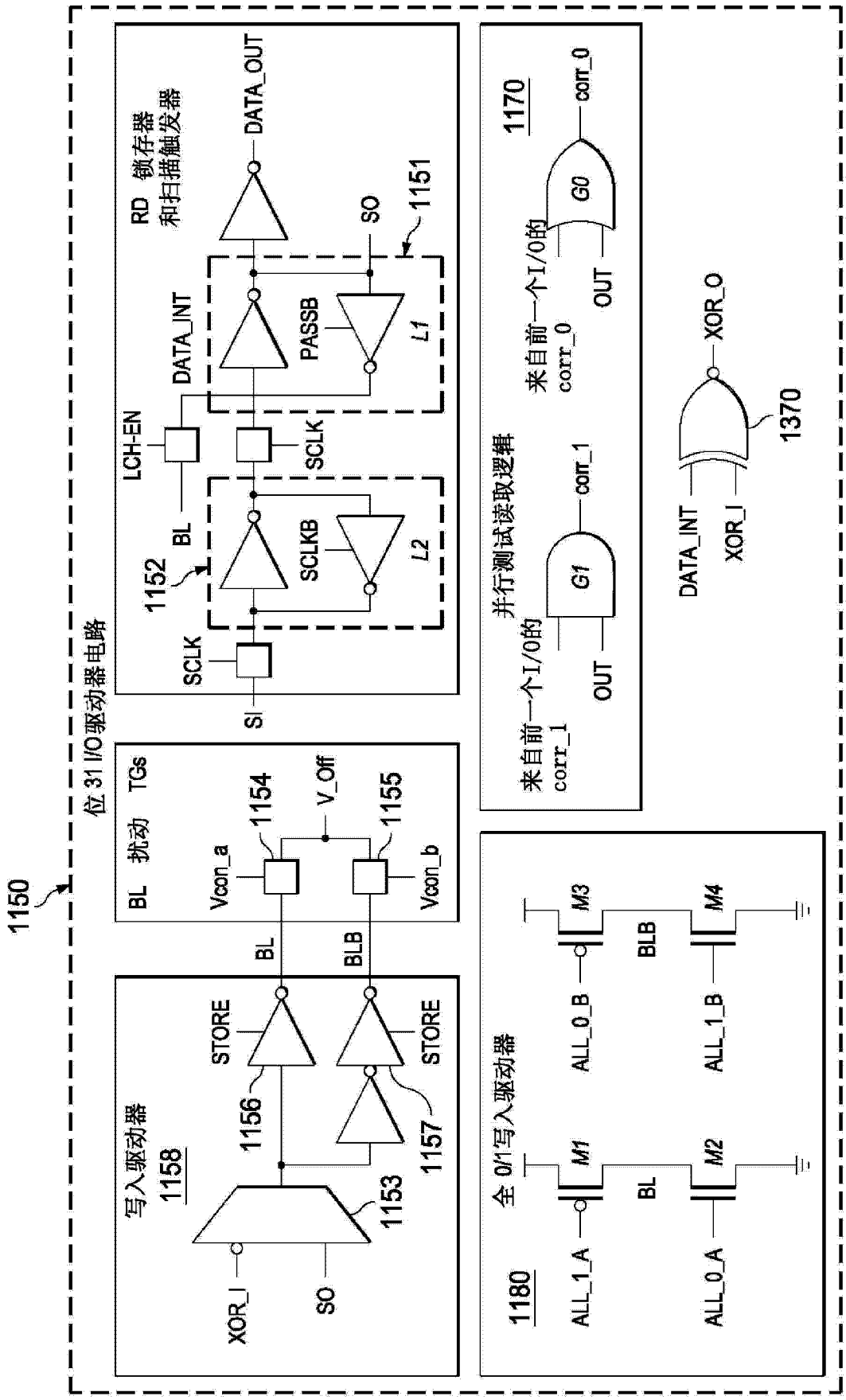

1150
位 31 I/O驱动器电路
RD 锁存器
和扫描触发器
DATA_OUT
1151
SO
SI
LCH-EN
DATA_INT
PASSB
L1
SCLK
BL
1152
SCLKB
L2
BL 扰动 TGs
Vcon_a
1154
V_Off
1155
Vcon_b
BL
BLB
写入驱动器
1158
STORE
1156
1157
1153
XOR_I
SO
并行测试读取逻辑
1170
来自前一个I/O的
corr_1
G1
corr_1
OUT
来自前一个I/O的
corr_0
G0
corr_0
OUT
DATA_INT
XOR_I
XOR_O
1370
全 0/1写入驱动器
1180
ALL_1_A
M1
ALL_0_B
M3
BL
BLB
ALL_0_A
M2
ALL_1_B
M4

图 11B

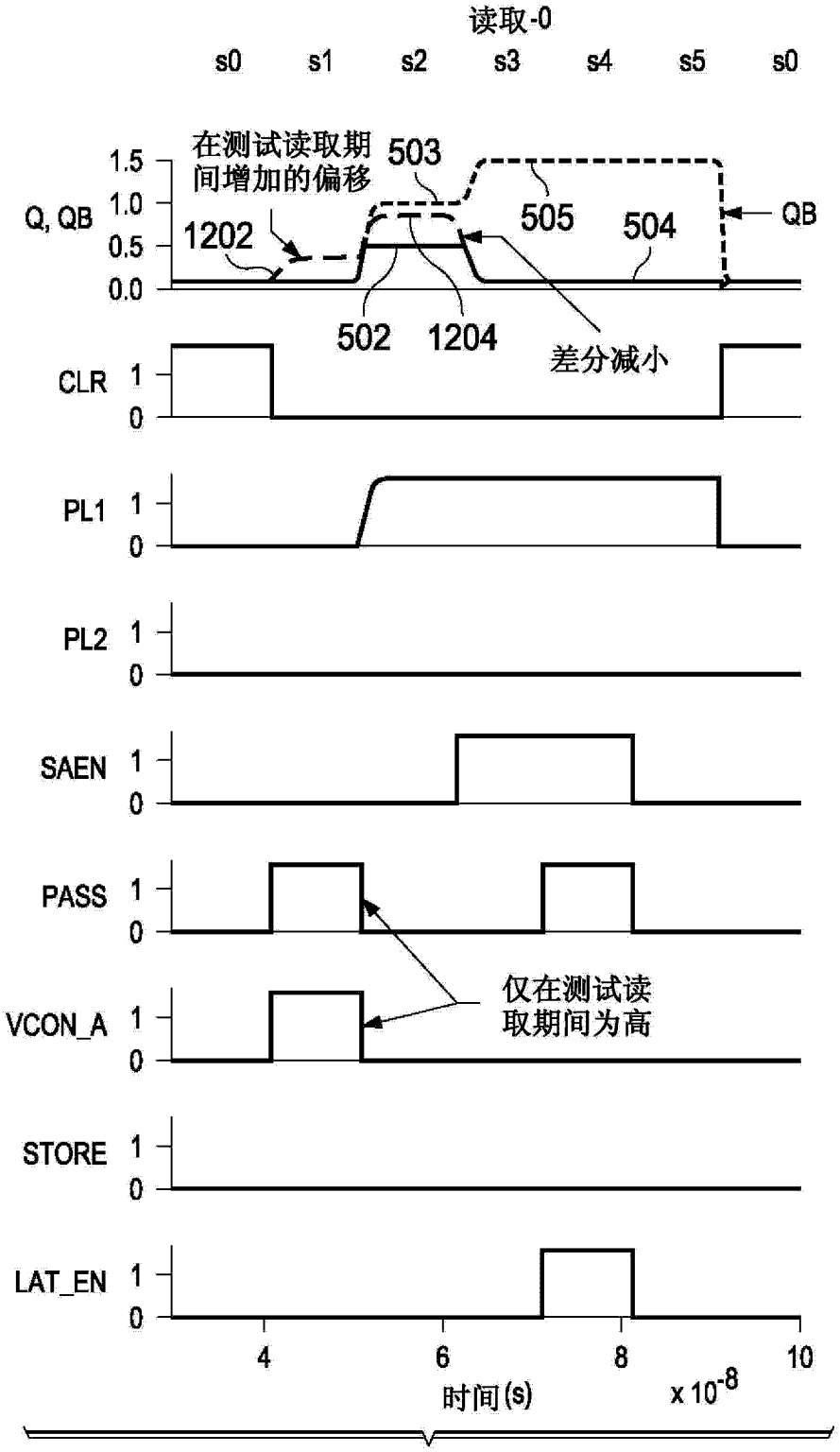

读取-0
s0
s1
s2
s3
s4
s5
s0
在测试读取期间增加的偏移
503
505
504
QB
Q, QB
1.5
1.0
0.5
0.0
1202
502
1204
差分减小
CLR
PL1
PL2
SAEN
PASS
仅在测试读取期间为高
VCON_A
STORE
LAT_EN
1
0
4
6
8
10
时间(s)
x 10-8

图 12A

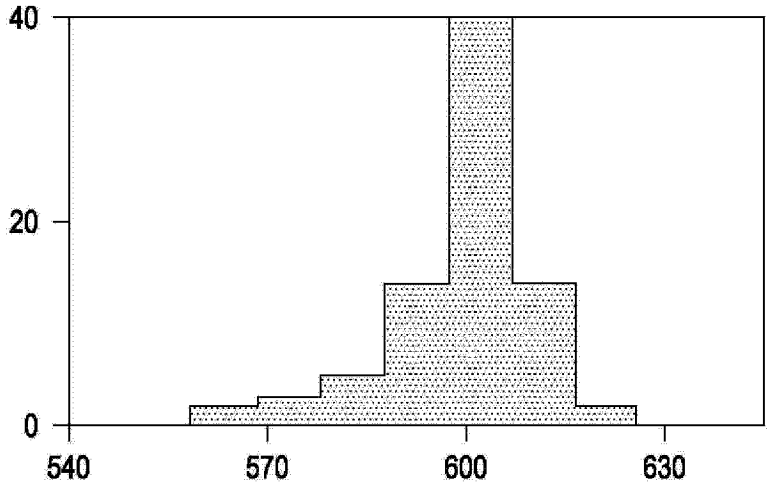


图 12B

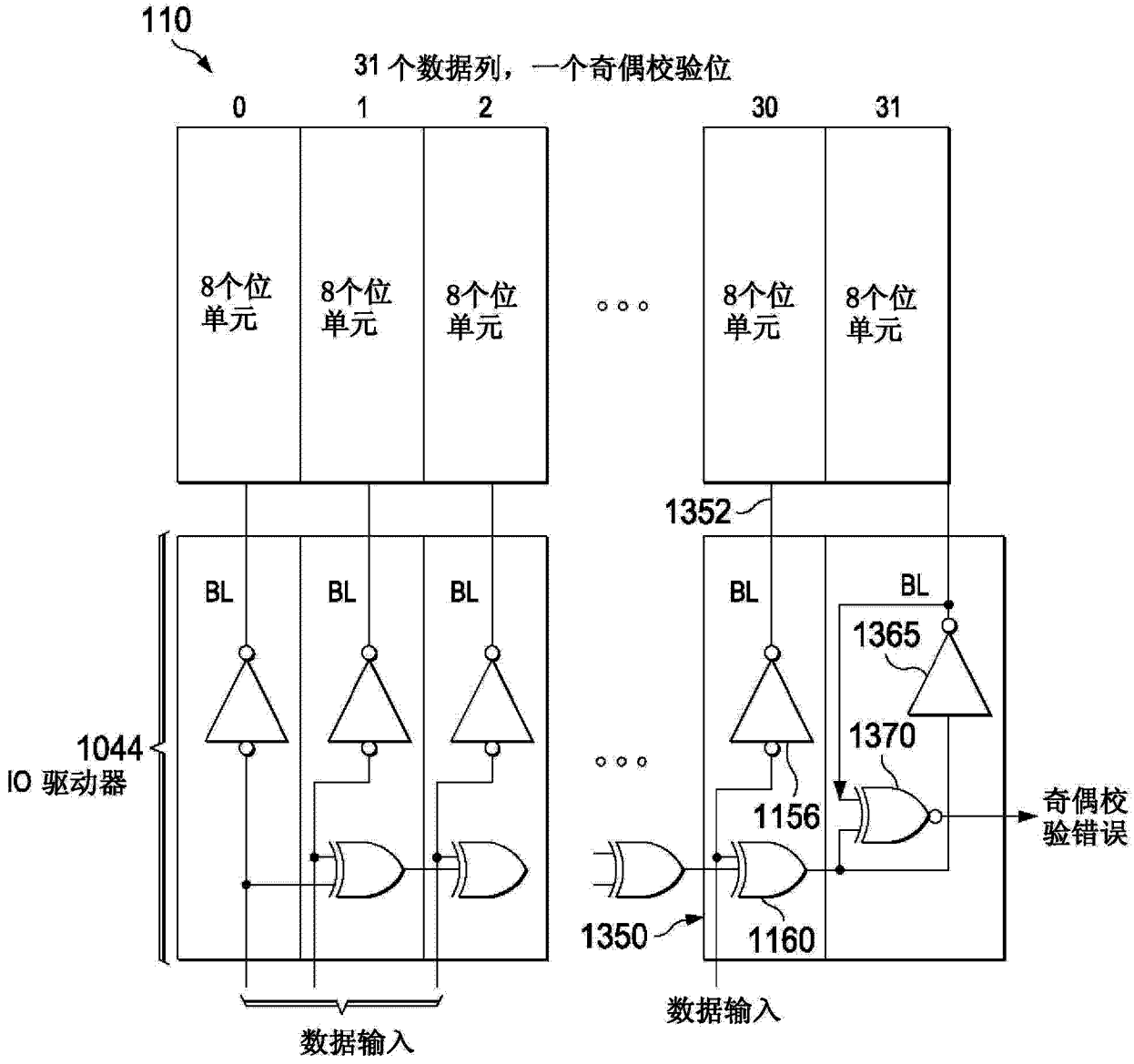

110
31 个数据列，一个奇偶校验位
0
1
2
30
31
8个位单元
8个位单元
8个位单元
8个位单元
8个位单元
1352
BL
BL
BL
BL
BL
1365
1370
1044
IO 驱动器
1156
奇偶校验错误
1350
1160
数据输入
数据输入

图 13

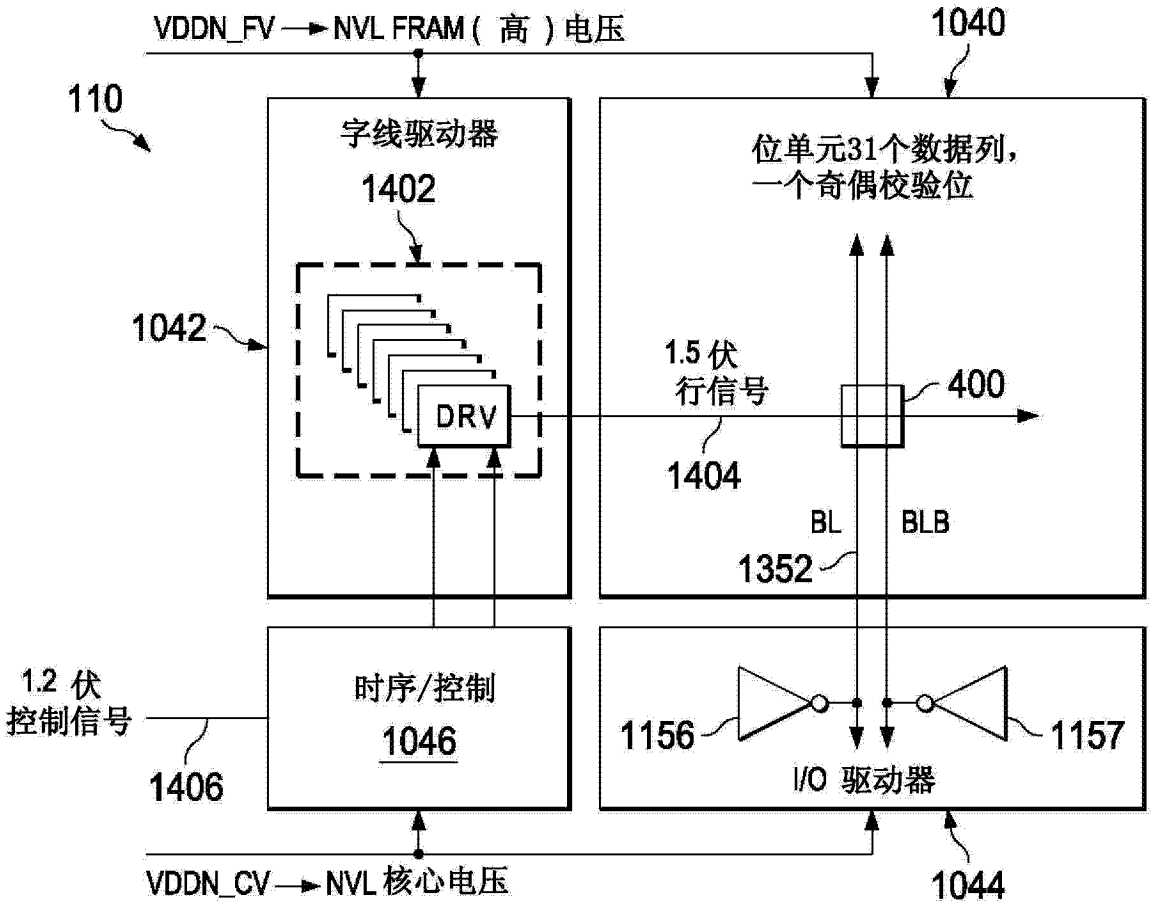

VDDN_FV
NVL FRAM (高) 电压
1040
110
字线驱动器
1402
位单元31个数据列，
一个奇偶校验位
1042
DRV
1.5 伏
行信号
400
1404
BL
BLB
1352
1.2 伏
控制信号
时序/控制
1046
1156
1157
I/O 驱动器
1406
VDDN_CV
NVL 核心电压
1044

图 14

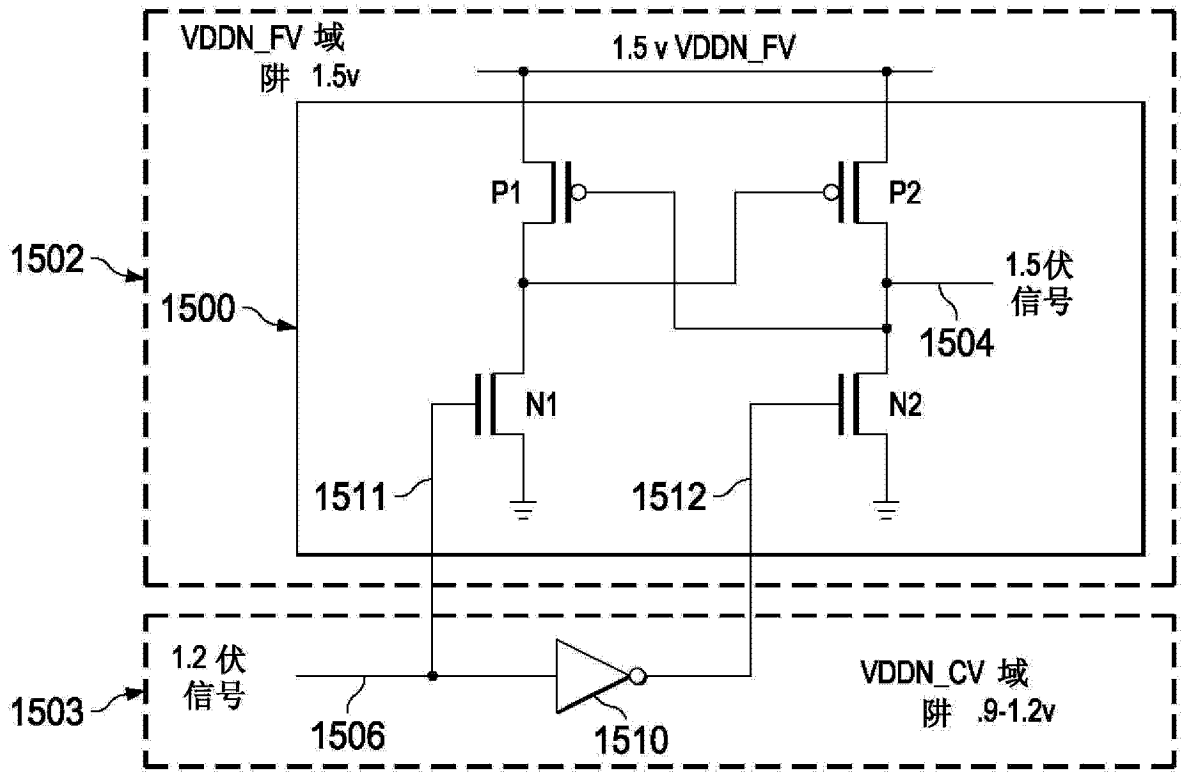

VDDN_FV 域
阱 1.5v
1.5 v VDDN_FV
P1
P2
1502
1500
1.5伏
信号
1504
N1
N2
1511
1512
1.2 伏
信号
1503
1506
1510
VDDN_CV 域
阱 .9-1.2v

图 15

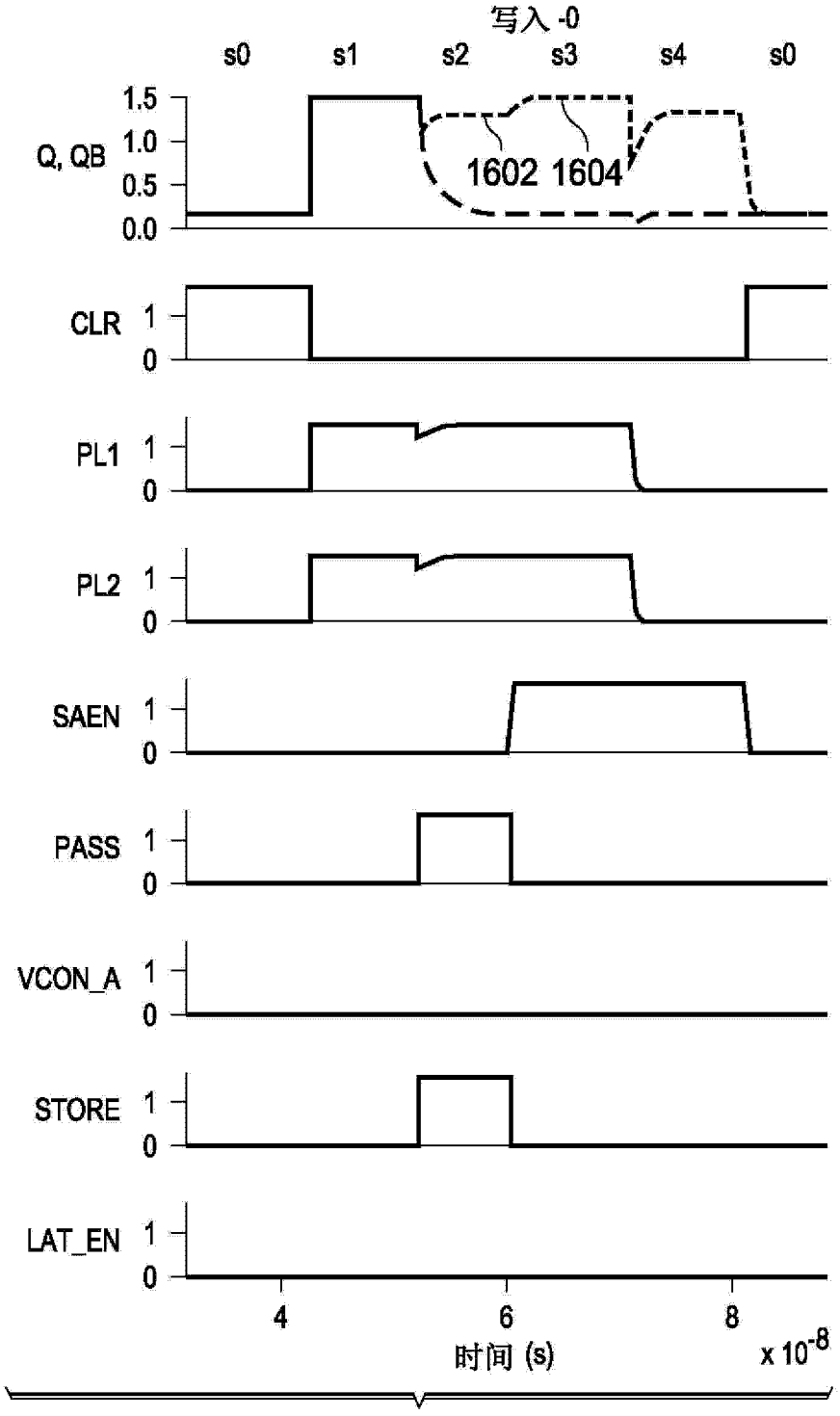

写入 -0
s0
s1
s2
s3
s4
s0
Q, QB
1.5
1.0
0.5
0.0
1602 1604
CLR
PL1
PL2
SAEN
PASS
VCON_A
STORE
LAT_EN
1
0
4
6
8
时间 (s)
x 10-8

图 16

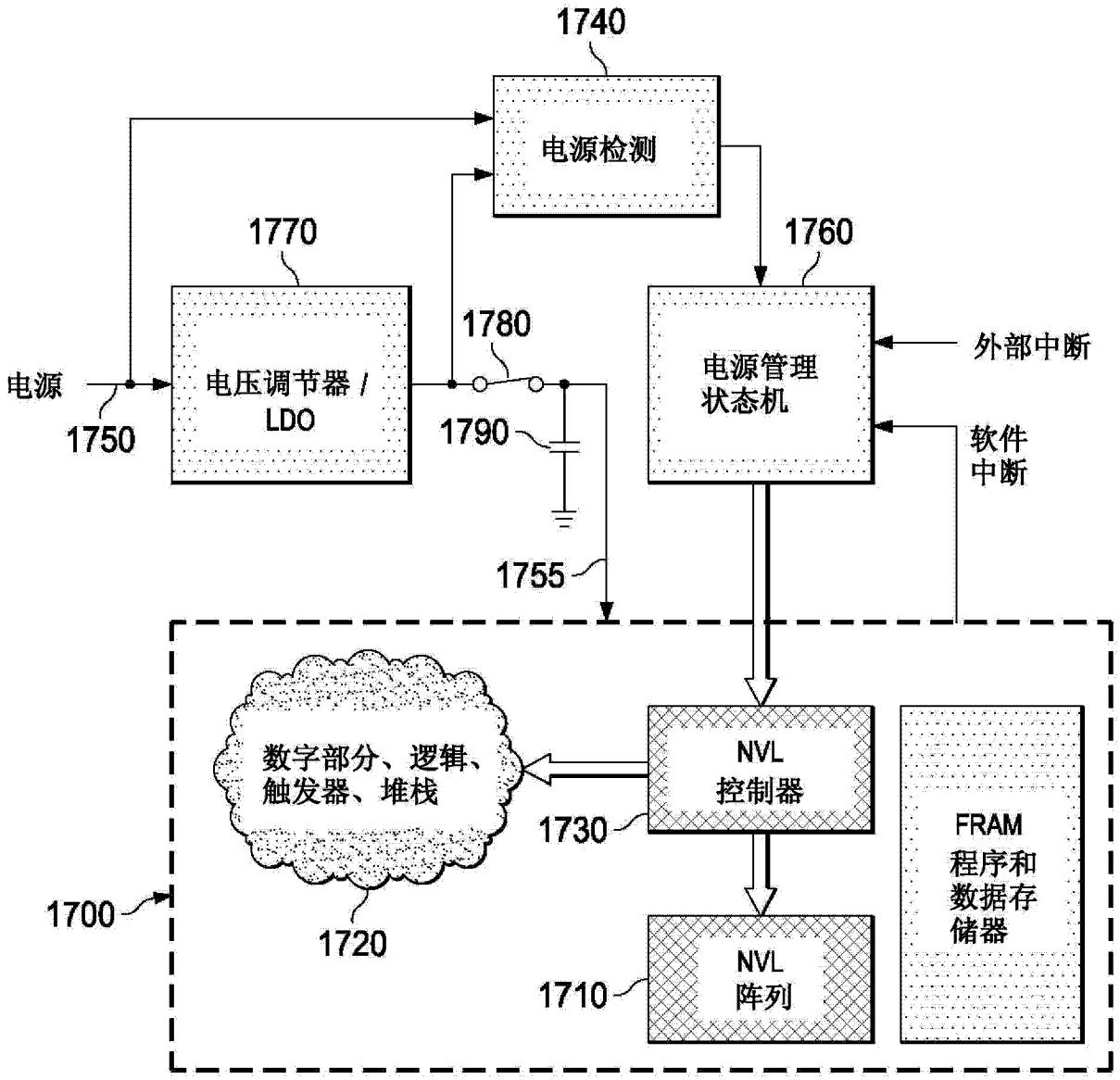

1740
电源检测
1770
1760
1780
电源
电压调节器 / LDO
电源管理状态机
外部中断
1750
1790
软件中断
1755
数字部分、逻辑、触发器、堆栈
NVL 控制器
FRAM 程序和数据存储器
1730
1700
1720
NVL 阵列
1710

图 17

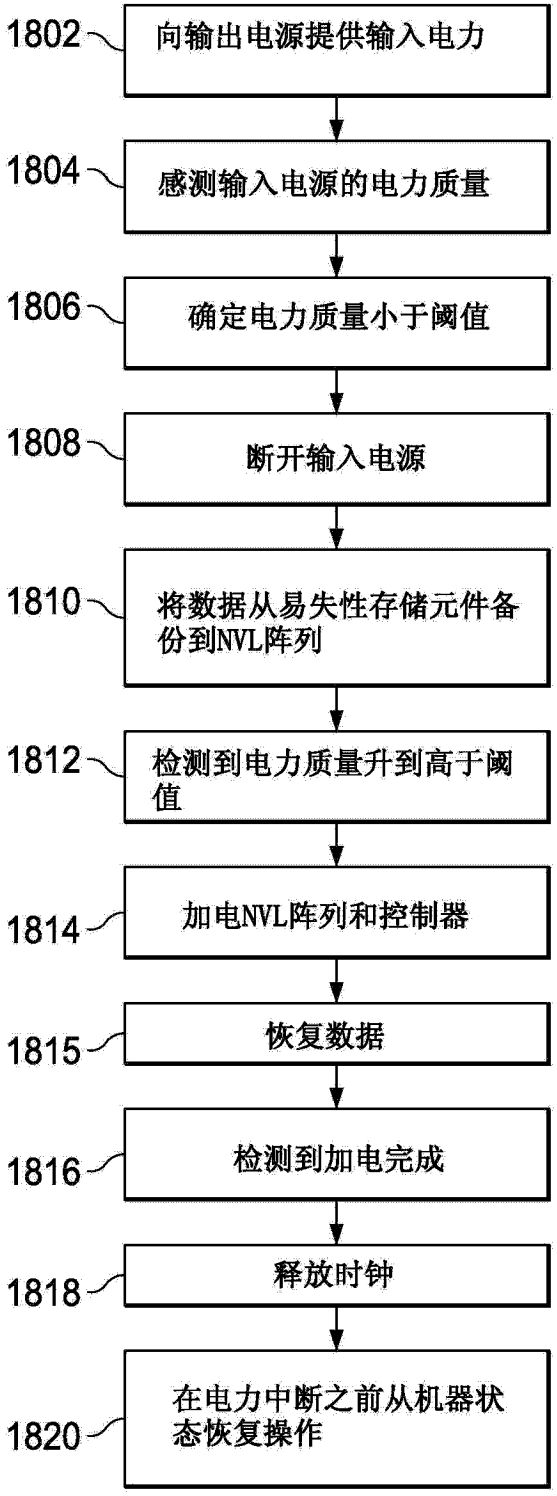

1802
向输出电源提供输入电力
1804
感测输入电源的电力质量
1806
确定电力质量小于阈值
1808
断开输入电源
1810
将数据从易失性存储元件备份到NVL阵列
1812
检测到电力质量升到高于阈值
1814
加电NVL阵列和控制器
1815
恢复数据
1816
检测到加电完成
1818
释放时钟
1820
在电力中断之前从机器状态恢复操作

图 18

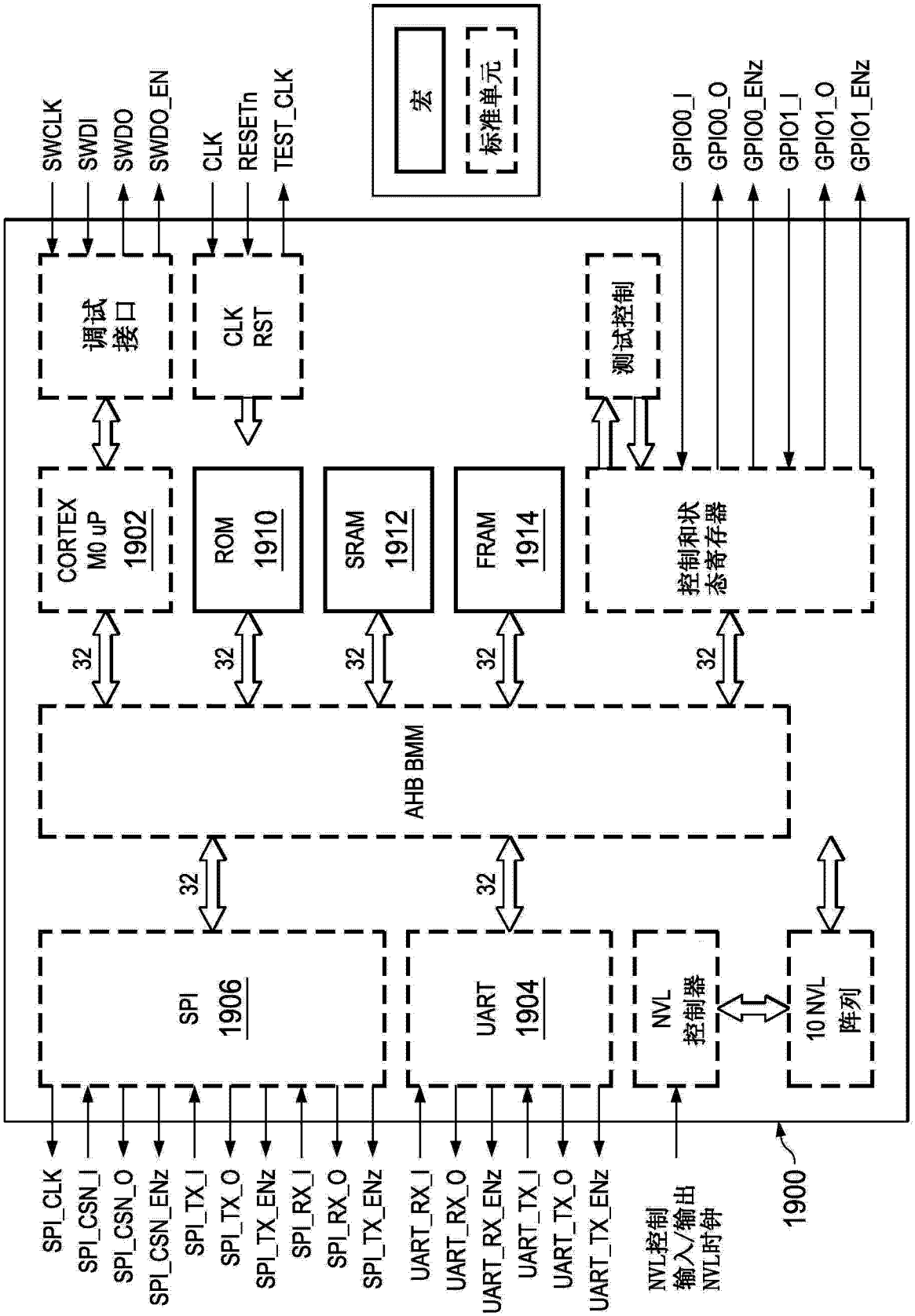

宏
标准单元
SWCLK
SWDI
SWDO
SWDO_EN
CLK
RESETn
TEST_CLK
GPIO0_I
GPIO0_O
GPIO0_ENz
GPIO1_I
GPIO1_O
GPIO1_ENz
调试接口
CLK RST
测试控制
CORTEX M0 uP 1902
ROM 1910
SRAM 1912
FRAM 1914
控制和状态寄存器
32
AHB BMM
SPI 1906
UART 1904
NVL 控制器
10 NVL 阵列
SPI_CLK
SPI_CSN_I
SPI_CSN_O
SPI_CSN_ENz
SPI_TX_I
SPI_TX_O
SPI_TX_ENz
SPI_RX_I
SPI_RX_O
SPI_TX_ENz
UART_RX_I
UART_RX_O
UART_RX_ENz
UART_TX_I
UART_TX_O
UART_TX_ENz
NVL控制输入/输出NVL时钟
1900

图 19