



(12)发明专利

(10)授权公告号 CN 106133710 B

(45)授权公告日 2019.10.11

(21)申请号 201580016122.2

(22)申请日 2015.06.23

(65)同一申请的已公布的文献号
申请公布号 CN 106133710 A

(43)申请公布日 2016.11.16

(30)优先权数据
14/318,114 2014.06.27 US

(85)PCT国际申请进入国家阶段日
2016.09.23

(86)PCT国际申请的申请数据
PCT/US2015/037210 2015.06.23

(87)PCT国际申请的公布数据
W02015/200338 EN 2015.12.30

(73)专利权人 超威半导体公司
地址 美国加利福尼亚州

(72)发明人 格伦·A·戴斯 格里·塔尔博特
安瓦尔·卡谢穆
爱德华多·普莱特
布莱恩·布莱恩

(74)专利代理机构 上海胜康律师事务所 31263
代理人 樊英如 张静

(51)Int.Cl.
G06F 13/40(2006.01)

(56)对比文件
CN 103502964 A, 2014.01.08,
CN 103168296 A, 2013.06.19,
US 2013155788 A1, 2013.06.20,
CN 103168296 A, 2013.06.19,
审查员 张桂华

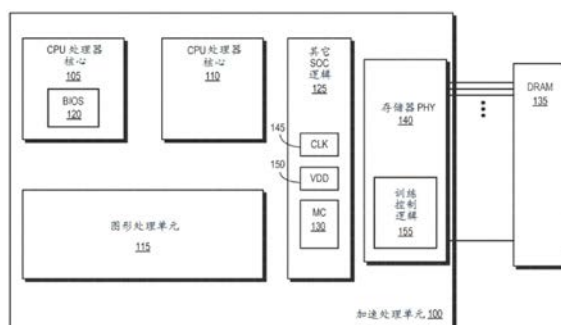
权利要求书2页 说明书9页 附图4页

(54)发明名称

用于训练存储器物理层接口的集成控制器

(57)摘要

集成于存储器物理层接口(PHY)(140、205)中的控制器(215)可用于控制训练,所述训练用于配置所述存储器PHY以便与诸如动态随机存取存储器(DRAM)等相关联的外部存储器(135)通信,由此不需要在BIOS(120)与所述存储器PHY之间的数据管道上提供训练序列。举例来说,集成于所述存储器PHY中的控制器(215)可基于训练算法来控制所述存储器PHY的读取训练和写入训练以便与所述外部存储器通信。所述训练算法可为无种训练算法,其收敛于所述存储器PHY与所述外部存储器之间的时序延迟和电压偏移的解,而不需要从所述BIOS接收表征由所述训练算法产生的训练序列或命令通过的信号路径的种子信息。



1. 一种用于训练存储器物理层接口的设备,其包括:

存储器物理层接口,其用于耦合到外部存储器,所述存储器物理层接口包括控制器,其中所述控制器将基于训练算法来控制所述存储器物理层接口的读取训练和写入训练以便与所述外部存储器通信;

第一训练引擎,其用于响应于由所述控制器基于所述训练算法产生的控制信令而产生全速可编程命令序列以便递送到所述外部存储器;

第二训练引擎,其用于响应于由所述第一训练引擎产生的控制信令而产生训练数据序列以便递送到所述外部存储器;以及

第三训练引擎,其用于将从所述外部存储器接收的序列与由所述第二训练引擎产生的所述训练序列进行比较。

2. 如权利要求1所述的设备,其中所述第一训练引擎基于所述训练算法对由所述第二训练引擎和所述第三训练引擎中的至少一者执行的任务的执行进行定序。

3. 如权利要求1所述的设备,其中所述第三训练引擎调节由所述存储器物理层接口使用的时序参数和电压偏移参数中的至少一者以便进行以下各者中的至少一者:从所述外部存储器读取数据;以及将数据写入到所述外部存储器。

4. 如权利要求3所述的设备,其中所述第三训练引擎基于取样训练数据中的正确样本和不正确样本的数目来确定所述时序参数和所述电压偏移参数中的至少一者。

5. 如权利要求1所述的设备,其中所述训练算法是无种训练算法。

6. 一种用于训练存储器物理层接口的方法,其包括:

在集成于存储器物理层接口中的控制器处基于训练算法而产生控制信令,以用于读取训练和写入训练所述存储器物理层接口以便与外部存储器通信;

将第一控制信令从所述控制器提供给第一训练引擎,其中所述第一训练引擎将响应于所述第一控制信令而产生全速命令序列以便递送到所述外部存储器;

将第二控制信令从所述第一训练引擎提供给第二训练引擎,其中所述第二训练引擎将响应于来自所述第一训练引擎的所述第二控制信令而产生训练序列以便递送到所述外部存储器;以及

将第三控制信令从所述第一训练引擎提供给第三训练引擎,其中所述第三训练引擎将响应于所述第三控制信令而将从所述外部存储器接收的序列与由所述第二训练引擎产生的所述训练序列进行比较。

7. 如权利要求6所述的方法,其进一步包括:

基于由所述第一训练引擎提供给所述第二训练引擎和所述第三训练引擎中的至少一者的信号,对由所述第二训练引擎和所述第三训练引擎中的所述至少一者执行的任务的执行进行定序。

8. 如权利要求6所述的方法,其进一步包括:

在所述第三训练引擎处确定由所述存储器物理层接口使用的时序延迟和电压偏移中的至少一者以便进行以下各者中的至少一者:从所述外部存储器读取数据;以及将数据写入到所述外部存储器。

9. 如权利要求8所述的方法,其中确定所述时序延迟和所述电压偏移中的所述至少一者包括:基于取样训练数据中的正确样本与不正确样本的预定比率来确定所述时序延迟和

所述电压偏移中的所述至少一者。

10. 如权利要求6所述的方法,其中产生所述控制信令包括:基于无种训练算法而产生所述控制信令。

11. 一种体现一组可执行指令的非暂时性计算机可读介质,所述一组可执行指令用于操纵至少一个处理器来:

在集成于存储器物理层接口中的控制器处基于训练算法而产生控制信令,以用于读取训练和写入训练所述存储器物理层接口以便与外部存储器通信,其中所述存储器物理层接口包括所述至少一个处理器,其中所述至少一个处理器用于基于无种训练算法而产生所述控制信令,以使得所述无种训练算法收敛于所述存储器物理层接口与所述外部存储器之间的时序延迟和电压偏移的解,而不需要从基本输入/输出系统 (BIOS) 接收表征由所述训练算法产生的训练序列或命令通过的信号路径的种子信息;

将第一控制信令从所述控制器提供给第一训练引擎,其中所述第一训练引擎将响应于所述第一控制信令而产生全速命令序列以便递送到所述外部存储器;

将第二控制信令从所述第一训练引擎提供给第二训练引擎,其中所述第二训练引擎将响应于来自所述第一训练引擎的所述第二控制信令而产生训练序列以便递送到所述外部存储器;以及

将第三控制信令从所述第一训练引擎提供给第三训练引擎,其中所述第三训练引擎将响应于所述第三控制信令而将从所述外部存储器接收的序列与由所述第二训练引擎产生的所述训练序列进行比较。

用于训练存储器物理层接口的集成控制器

[0001] 背景

[0002] 本公开的领域

[0003] 本申请大体上涉及处理系统,且更特定来说,涉及处理系统中的存储器物理层接口。

[0004] 相关技术的描述

[0005] 例如系统芯片(SOC)等处理系统使用存储器来存储数据或指令以供稍后使用。举例来说,SOC可包括处理单元,例如中央处理单元(CPU)、图形处理单元(GPU),且加速处理单元(APU)可从存储器读取指令或数据、使用所述指令或数据执行操作,且随后将结果写回到存储器中。处理系统可包括存储器物理层接口以用于控制对例如动态随机存取存储器(DRAM)等存储器模块的存取,所述存储器模块可用于存储信息,使得所存储的信息可由处理单元在处理系统的操作期间存取。处理系统中的存储器物理层接口常规上被称作“存储器PHY”。存储器控制器通常用于控制存储器PHY的操作。

[0006] 在可从DRAM准确地读取数据或将数据准确地写入到DRAM之前,通常使用在存储器PHY与DRAM之间的接口上交换的序列来训练存储器PHY。接口训练程序可包括全局接口初始化机制以便在读取和写入的接口端点处执行命令发出和数据样本时序的相对粗粒度的同步。全局同步的训练序列可(例如)使用以周期性方式被命令间隙(其可被称作“气泡”)打断的DRAM命令的连续流构建。此序列在连续流内创建一连串时间/事件标记,其可用于建构粗糙/全局接口时序或使其同步。更细粒度的每个信号训练程序随后可用于找到用于对数据信号进行取样的最佳时序和电压偏移。每个信号训练程序的训练序列可包括从DRAM读取和/或写入到DRAM的“1”和“0”的随机混合。响应于训练序列而从DRAM返回的信号用于确定“数据眼”的轮廓,其为展示表示作为时间和电压的函数的数据信号的值的上升沿、下降沿和高/低电压的曲线图。

[0007] 通过在处理系统中的处理器上运行的BIOS代码起始常规的训练程序。然而,BIOS通常在逻辑上或物理上远离存储器PHY。举例来说,BIOS和存储器PHY可被数据管道、高速缓冲存储器和缓冲器分离。这些介入的接口和子系统块可能会扭曲或更改从BIOS传输的训练控制和数据序列,例如,可能将额外的气泡引入到训练序列中,从而使其难以找到有效的训练算法。因此,BIOS通常将复杂的训练序列播种和数据后处理方案并入到训练代码中。这两种方法都具有显著的缺陷。举例来说,在后处理期间移动、开采数据且对数据操作所需的额外时间会增加执行训练所需的总时间。如果总训练时间预算是固定的,那么花在后处理的额外时间意味着必须在其它地方节约时间以将总训练时间保持在预算内。可通过运行较短的模式来节约时间,其可导致较大的训练误差。对于另一实例,训练序列播种用于从系统状态开始训练,其中避免了上文所描述的可能会阻止算法收敛的管道问题,但找到适当的种子值来正确地界定初始系统状态需要密集系统和平台表征。所述种子值可包括表征以下各者的信息:数据管道、存储器PHY或沿着从BIOS穿过存储器PHY到DRAM且回到BIOS的路径的其它逻辑或物理实体。必须在实验室针对每个处理系统或DRAM的实施表征种子值中的每一者。因此,随着处理系统和相关联的存储器装置的数目和多样性增加,必须产生的种子信息

的数量和种类会快速扩大,这可能会增大BIOS代码大小且使得BIOS更为难以维持。

[0008] 附图简述

[0009] 通过参考附图,可更好地理解本公开,且使其众多特征和优势对于本领域技术人员显而易见。在不同图式中使用相同的参考符号会指示类似或等同的项目。

[0010] 图1是根据一些实施方案的包括一个或多个中央处理单元(CPU)处理器核心和图形处理单元(GPU)的加速处理单元(APU)的框图。

[0011] 图2是根据一些实施方案的包括存储器PHY和DRAM的处理单元的一部分的框图。

[0012] 图3是根据一些实施方案的可在例如图1中所示的加速处理单元等处理单元中实施的训练存储器PHY的方法的流程图。

[0013] 图4是示出根据一些实施方案的用于设计和制造实施处理系统的组件的至少一部分的集成电路装置的方法的流程图。

[0014] 详细描述

[0015] 如本文论述,BIOS与存储器PHY之间的数据管道可使得更难以找到可在BIOS中实施来控制提供给DRAM的训练序列的时序的有效训练算法。此外,训练序列中的改变可使得BIOS中的训练算法更难以处理DRAM返回的信号。将训练算法从BIOS移动到存储器PHY可减少可能由后处理引入的系统时序预算上的压力,执行所述后处理来补偿由数据管道导致的训练序列中的失真,进而提高对时序序列的时序的控制,且在一些实施方案中,减少或消除对播种训练算法的需要。在存储器PHY中实施训练算法还可简化用于实施存储器PHY的硬件的设计,且允许工程师创建根据处理系统而调谐的更有效的训练算法。在硬件中实施训练算法的实施方案(例如,作为用于实施存储器PHY的硬件的部分)还可允许“全速”执行测试,这意味着可以DRAM接口的最大速率传输命令和训练序列。

[0016] 在一些实施方案中,存储器PHY内的例如微控制器等集成控制器可用于控制训练算法,所述训练算法用于配置所述存储器PHY以便与相关联的DRAM通信,进而不需要在BIOS与存储器PHY之间的数据管道上提供训练序列。训练算法因此可实施为无种训练算法,因为在沿着数据管道传达训练序列时不需要考虑训练序列的失真或更改。如本文所使用,“种子”信息是指表征由训练算法产生的训练序列或命令通过的信号路径的信息。训练算法使用种子信息来建立初始系统状态,使得训练算法可收敛于同步解,例如,使得对所返回信号的后处理可收敛于对存储器PHY与相关联的DRAM之间的相位差(或时序延迟)与电压偏移的解。“播了种的”训练算法因此需要种子信息来进行收敛。相比而言,“无种”训练算法可在不使用种子信息建立初始状态的情况下收敛于同步解。然而,在一些实施方案中,BIOS仍可提供可由无种训练算法使用的配置信息,例如识别DRAM的配置的信息,但从BIOS接收此信息对于确保收敛可能不是必需的。

[0017] 在一些实施方案中,微控制器耦合到:第一训练引擎,其产生被递送到DRAM的可编程命令序列;第二训练引擎,其产生训练序列;以及第三训练引擎,其基于提供给DRAM的训练序列而将从DRAM接收的业务与其预期值进行比较。举例来说,所述第三训练引擎可使一个或多个所传输的训练序列与所接收的信号相关以确定所传输的序列与所接收的序列之间的时序延迟。第三训练引擎还可调节由存储器PHY使用的时序或电压偏移参数。第一训练引擎可对由第二和第三训练引擎执行的任务的执行进行定序。

[0018] 图1是根据一些实施方案的包括一个或多个中央处理单元(CPU) 105、110处理器核

心和图形处理单元 (GPU) 115 的加速处理单元 (APU) 100 的框图。APU 100 可实施为系统芯片 (SOC)。CPU 处理器核心 105、110 可独立地、同时地或并行地执行指令。虽然图 1 中所示的 APU 100 包括两个 CPU 处理器核心 105、110, 但受益于本公开的本领域普通技术人员应了解, APU 100 中的处理器核心的数目与设计选择有关。APU 100 的一些实施方案可包括多于或少于图 1 中所示的两个 CPU 处理器核心 105、110。GPU 115 用于创建既定用于输出到显示器的视觉图像。GPU 115 的一些实施方案还可包括多个处理核心 (未示出)。

[0019] CPU 处理器核心 105 包括基本输入/输出系统 (BIOS) 120, 其可实施于硬件、固件、软件或其组合中。BIOS 120 的一些实施方案用于 (例如) 响应于包括 APU 100 的系统被通电或启动而初始化或测试 APU 100 的组件。BIOS 120 还可用于加载操作系统。可使用一个或多个数据管道 (图 1 中未示出) 将由 BIOS 120 产生的指令或命令传达到 APU 100 中的其它位置。BIOS 120 可替代地实施于 CPU 处理器核心 110 或 APU 100 内的其它位置中。

[0020] 图 1 中所示的 APU 100 还包括其它 SOC 逻辑 125。其它 SOC 逻辑 125 的一些实施方案可包括 DMA 引擎 (未示出), 以便产生地址且初始化存储器读取或写入循环、以便执行存储器间的数据传递, 或以便在 CPU 处理器核心 105、110 或 GPU 115 之间传递数据。其它 SOC 逻辑 125 还可包括路由逻辑、相干逻辑, 或用于实施其它功能性的逻辑。其它 SOC 逻辑 125 的一些实施方案包括存储器控制器 (MC) 130 来协调 APU 100 与其它存储器 (例如, 外部动态随机存取存储器 (DRAM) 135) 之间的数据流。存储器控制器 130 包括用于控制从外部存储器读取信息以及将信息写入到外部存储器的逻辑。存储器控制器 130 还可包括刷新逻辑, 其用于将信息周期性地重新写入到 DRAM, 使得保持 DRAM 的存储器单元中的信息。DRAM 的一些实施方案可为双倍数据速率 (DDR) DRAM, 在此情况下, 存储器控制器 130 可能能够在存储器时钟的上升沿和下降沿两者上传递去往和来自 DRAM 的数据。

[0021] 存储器控制器 130 可使用经由存储器物理层接口 140 (其可被称作存储器 PHY 140) 传输的信号来控制例如 DRAM 135 等其它存储器模块的操作。存储器 PHY 140 包括用于驱动信号的电路, 所述信号操纵可耦合到 APU 100 的其它存储器模块的操作。举例来说, 存储器 PHY 140 可提供控制例如 DRAM 135 等存储器模块的读取、写入、刷新或擦除部分的信号。存储器 PHY 140 可能能够在不同操作点处操作, 所述操作点可由存储器 PHY 140 的操作频率和/或操作电压确定。举例来说, 其它 SOC 逻辑 125 可包括时钟 145, 其提供用于操纵存储器 PHY 140 和/或存储器控制器 130 和参考电压 (VDD) 150 中的同步的时钟信号, 所述参考电压操纵由存储器 PHY 140 和/或存储器控制器 130 使用的电压。

[0022] 应该训练存储器 PHY 140 以便提高存储器 PHY 140 与 DRAM 135 之间的通信期间的读取或写入性能。存储器 PHY 140 因此包括集成的训练控制逻辑 155, 其用于: 产生训练序列或命令; 将训练序列或命令传输到 DRAM 135; 响应于传输的训练或命令而接收由 DRAM 135 产生的信号; 以及基于来自 DRAM 135 的响应而调节存储器 PHY 140 的读取/写入参数。

[0023] 将训练控制逻辑 155 集成到存储器 PHY 140 中与使用 BIOS 120 中实施的算法来训练存储器 PHY 140 的常规实践相比具有若干优势。由于不需要在 BIOS 120 与存储器 PHY 140 之间的数据管道上传输训练序列, 所以可减少或消除对由训练控制逻辑 155 使用的训练算法的后处理和/或播种。此外, 将支持存储器 PHY 140 的训练的训练控制逻辑 155 并入到存储器 PHY 140 中显著没有设计 BIOS 120 来支持存储器 PHY 140 的训练那么复杂。举例来说, 为了设计 BIOS 训练算法, 必须向负责设计 BIOS 训练算法的工程师导出过多的复杂存储器 PHY

初始化和控制细节。举例来说,存储器PHY 140可包括用于界定时序、锁定频率等的数百个寄存器(出于清楚起见在图1中未示出),且这些寄存器中的每一者需要以正确序列初始化和存取。BIOS工程师可能不熟悉存储器PHY 140的操作,或至少显著没有负责设计存储器PHY 140的工程师那么熟悉存储器PHY 140的操作。将设计存储器PHY训练算法的责任放到存储器PHY工程师的手上可因此导致更有效的设计,这可进一步减少对训练算法的播种的需要。另外,对多个PHY例子的训练必须序列化,且BIOS资格与新训练代码的发布和种子信息的实验室表征紧密联系。此外,在使用BIOS训练算法来训练存储器PHY 140之前,APU 100中的许多块可能需要经过配置且运行来支持BIOS 120与存储器PHY 140之间的数据管道,这可能会增加每个目标平台的上市时间。

[0024] 图2是根据一些实施方案的包括存储器PHY 205和DRAM210(例如,图1中所示的存储器PHY 140和DRAM 135)的处理单元的一部分200的框图。存储器PHY 205包括控制器215,所述控制器用于基于训练算法来控制存储器PHY 205的读取训练和写入训练以便与DRAM 210通信。控制器215的一些实施方案可实施于硬件、固件、软件或其组合中。在一些实施方案中,控制器215还可被称作微控制器215。由于控制器215被集成到存储器PHY 205中且因此未通过数据管道与存储器PHY 205分开,所以由控制器215实施的训练算法可为无种训练算法。如本文论述,不需要使用表征由训练算法产生的训练序列或命令通过的信号路径或数据管道的信息来配置无种训练算法(例如,用于对所返回的信号的后处理)。控制器215可响应于上电、垂直回扫频率变化、功率状态转变(例如,在闲置与活动状态之间)、以预定频率周期性地,或响应于其它事件或信号,而起始训练。

[0025] 控制器215可与BIOS(例如,图1中所示的BIOS 120)交互。在一些实施方案中,BIOS可通过加载存储器阵列(例如,本地静态随机存取存储器(SRAM))而配置控制器215,从而提供与DRAM 210相关联的配置信息,或响应于系统启动而指令控制器215起始训练序列。然而,一旦已经配置控制器215,控制器215的一些实施方案基本上与BIOS独立地且基本上在没有来自BIOS的输入的情况下控制存储器PHY 205的训练。举例来说,控制器215实施训练算法,所述训练算法与BIOS独立地操作且可在不接收来自BIOS的训练命令或训练序列的情况下训练存储器PHY 205。由于控制器215可在没有来自BIOS的输入(其将必须在数据管道上从BIOS传输到控制器215)的情况下控制存储器PHY 205的训练,所以可将训练算法实施为无种训练算法。

[0026] 控制器215耦合到第一训练引擎220,其还可被称作地址命令状态机(ACSM) 220。ACSM 220产生命令,所述命令可在存储器PHY205的训练期间提供给DRAM 210。对于实施于硬件中以作为存储器PHY 205的集成部分的第一训练引擎220的实施方案,可“全速地”产生可编程命令。所述命令可以按由控制器215确定的序列存储在寄存器(未示出)中。随后可响应于控制器215将开始位写入到ACSM 220中的控制寄存器中而向DRAM 210发出所述命令。由ACSM 220产生的命令可包括用于从DRAM 210中的指定位置读取信息的读取命令,和用于将信息写入到DRAM 210中的指定位置的写入命令。ACSM 220的一些实施方案可产生将并发的读取命令和写入命令进行组合的环回命令,其将信号驱动到存储器PHY 205的物理引脚,所述信号随后沿着穿过存储器PHY 205的路径返回。环回命令因此可用于测试存储器PHY 205,而不需要DRAM 210连接到存储器PHY205的物理引脚。ACSM 220的一些实施方案可产生循环命令,所述循环命令在命令之间的指定延迟下反复地执行一个或多个命令,从而在执

行期间循环或重复单个指令、以序列循环经过多个命令,和类似者。

[0027] 控制器215还耦合到第二训练引擎225,其可被称作PRBS模式产生器检查器 (PPGC) 225。PPGC 225的一些实施方案是可编程的,且可产生用作用于训练存储器PHY 205的训练序列的数据流。举例来说,PPGC 225可响应于由控制器215提供的信令而产生用于任何16位或更少位的多项式的数据流。PPGC 225的一些实施方案包括用于产生训练序列的单独产生器235,和用于检查在存储器PHY 205与DRAM 210之间流动的包括训练序列的读取流或写入流的同步的检查器230。PPGC 225的操作可由从ACSM 220接收的信令控制。举例来说,ACSM 220可提供对操作(例如,在产生器235处产生训练序列)的执行进行定序的信令。

[0028] 控制器215还耦合到第三训练引擎,其可被称作数据训练状态机 (DTSM) 240。DTSM 240将从DRAM 210接收的业务与提供给DRAM210的训练序列进行比较,以便确定是否调节由存储器PHY 205使用的时序参数或电压偏移参数。举例来说,PPGC 225可将训练序列的表示提供给DTSM 240以便与在存储器PHY 205的读取训练或写入训练期间从DRAM 210返回的序列进行比较。在开始训练循环之前,控制器215可配置DTSM 240来控制由存储器PHY 205使用的时序参数或电压偏移参数。控制器215随后可对ACSM 220和PPGC 225进行编程来驱动一个或多个训练序列。DTSM 240随后可将由PPGC225产生的训练序列与已经从DRAM 210接收的序列进行比较。举例来说,DTSM 240可使训练序列与处于多个不同延迟的接收序列相关。基于所述比较,DTSM 240决定是否例如通过递增或递减这些参数中的一或多者而调节时序参数或电压偏移参数。举例来说,可基于根据训练序列与接收序列的相关度而确定的延迟来增加或减小时序偏移。DTSM 240的一些实施方案还可实施具有上限阈值或下限阈值比较逻辑的数据过滤器或二进制加法器以便训练到数据轮廓眼位置。

[0029] 可使用多组先入先出 (FIFO) 缓冲器在将训练序列提供给DRAM210之前对训练序列进行缓冲,且在从DRAM 210接收到所接收的序列之后对所接收的序列进行缓冲。举例来说,可使用一组出站FIFO缓冲器245对出站业务进行缓冲,且可使用一组进站FIFO缓冲器250对进站业务进行缓冲。可使用一个或多个接收器255在通道上将信号接收到DRAM 210,且将它们提供给进站FIFO缓冲器250。可使用一个或多个驱动器260、265在所述通道上将来自出站FIFO缓冲器245的信号传输到DRAM 210。举例来说,可使用驱动器260将数据 (DQ) 或时序 (DQS) 信号驱动到通道270上,且接收器255可在通道270上接收数据 (DQ) 或时序 (DQS) 信号。对于另一实例,可使用驱动器265在通道275上将地址 (ADDR) 或命令 (CMD) 驱动到DRAM 210。可调节由接收器255或驱动器260、265使用的时序延迟和电压偏移。

[0030] 存储器PHY 205包括时序/电压控制逻辑280。DTSM 240可将信号提供给时序/电压控制逻辑280来指示对时序参数的调节。举例来说,DTSM 240可基于提供给DRAM 210的训练序列与从DRAM 210接收的序列的比较而指令时序/电压控制逻辑280递增或递减时序延迟或电压偏移。时序/电压控制逻辑280随后可将控制信号提供给接收器255或驱动器260、265来调节由接收器255或驱动器260、265使用的时序延迟或电压偏移。时序/电压控制逻辑280的一些实施方案可用于调节多个级中的时序延迟或电压偏移,所述多个级例如为接收启用级、写入平衡级、读取训练级、写入训练级,和用于确定存储器PHY 205与DRAM 210之间的接口的数据眼轮廓的电压电平的级。

[0031] 图3是根据一些实施方案的可在例如图1中所示的加速处理单元100等处理单元中实施的训练存储器PHY的方法300的流程图。方法300的实施方案可实施于训练控制逻辑中,

例如图1中所示的训练控制逻辑155、或图2中所示的控制器215、ACSM 220、PPGC 225、DTSM 240和时序/电压控制逻辑280。

[0032] 在框305处,所述训练控制逻辑执行接收启用训练来确定何时启用存储器PHY在与DRAM的接口上接收数据。一些实施方案通过传输用于从DRAM读取选定地址的读取命令而执行接收启用训练。使用气泡序列穿插所述读取命令,所述气泡在从DRAM接收的信号中产生气泡的对应序列。训练控制逻辑随后监视从DRAM接收的信号,以便使存储器PHY中的命令产生的时间与来自DRAM的命令响应返回到存储器PHY的时间对准。将气泡间距时间间隔范围的大小设定成大于最差情况往返等待时间加上路径中的任何内部存储器PHY和DRAM等待时间。这避免了命令响应关联与较早或较晚响应的任何潜在混叠。存储器PHY可被配置成在接收启用训练级期间保持在连续读取状态中。举例来说,控制器215可配置ACSM 220、PPGC 225和DTSM 240且随后起始训练级。ACSM 220可发出命令/地址以便将训练序列写入到DRAM 210,且随后发出命令/地址以便从DRAM 210读取回训练序列。在一些实施方案中,实际上响应于所发出的命令没有将信息写入到DRAM 210,且忽视了DQ总线。仅监视返回的DQS。所发出的命令因此类似于读取命令,但DTSM 240不关心响应于所述命令返回了何种数据。DTSM 240仅关注于调节从DRAM 210回来的DQS选通的时序。训练序列可由PPGC 225产生且提供给DRAM 210。DTSM 240随后可使来自DRAM 210的接收数据与训练序列相关以便识别往返延迟,且指令时序/电压控制逻辑280调谐适当的接收器/驱动器(例如,接收器255和驱动器260、265)的参数,以便清除检测到的往返延迟。

[0033] 在框310处,训练逻辑执行写入平衡以将存储器PHY所使用的时钟信号对准到DRAM所使用的时钟信号。训练逻辑的一些实施方案因此可传输存储器PHY时钟信号和时序(DQS)信号,其用于对DRAM处的时钟的值进行取样。训练逻辑随后可使用在DQ总线上返回的DRAM时钟的取样值(例如)通过引入延迟来使DQS信号与DRAM内部的存储器时钟相位对准,而使存储器PHY时钟与DRAM时钟对准。举例来说,响应于来自控制器215的信令,ACSM 220可产生写入命令,所述写入命令致使包括上升沿的存储器PHY时钟信号与提供给DRAM 210的DQS信号对DRAM中的存储器时钟进行取样。DRAM时钟的取样值随后可返回到存储器PHY 205。PPGC 225中的检查器230产生内部比较值且将此值提供给DTSM 240。DTSM 240随后可将所述内部比较值与从DRAM 210接收的取样时钟信号值进行比较,且基于所述比较而产生调节信号以使写入DQS对准到DRAM 210中的时钟。DTSM 240随后可指令时序/电压控制逻辑280调谐接收器255和驱动器260、265的时序参数以便使存储器PHY时钟和DRAM时钟同步。举例来说,如果内部比较值是“0”且DRAM时钟的取样值是“1”,那么DTSM 240可指令时序/电压控制逻辑280将存储器PHY 205的时序提前预定时间量。如果内部比较值是“1”且DRAM时钟的取样值是“1”,那么DTSM 240可指令时序/电压控制逻辑282将存储器PHY 205的时序延迟预定时间量。此过程可迭代以便将存储器PHY时钟和DRAM时钟的同步调谐到预定容限内。

[0034] 在框315处,训练逻辑执行读取/写入相位训练以便基于存储器PHY与DRAM之间的读取/写入数据路径来确定数据眼轮廓的一维时间边界。训练逻辑的一些实施方案可因此传输一连串命令以便将训练序列写入到DRAM中的地址中,且随后以不同延迟循环读取DRAM的所寻址的序列之外的训练序列,以便确定数据眼轮廓的一维时间边界。举例来说,响应于来自控制器215的信令,ACSM 220可发出命令以便将由PPGC 225产生的一个或多个序列写入到DRAM 210中的一个或多个地址。ACSM 220随后可将一连串读取命令发出到以不同的延

迟值循环的DRAM 210中的地址。DTSM 240随后可将所循环的读取命令中的每一者的所接收的序列与所提供的训练序列进行比较以便确定数据眼轮廓的左边缘和右边缘。DTSM 240随后可指令时序/电压控制逻辑280调谐接收器255的时序参数(例如,相位),从而对应于数据眼轮廓中的预定位置,例如左边缘与右边缘之间的中点。

[0035] 在框320处,训练逻辑执行二维(2D)读取/写入相位训练以便基于存储器PHY与DRAM之间的读取/写入数据路径来确定数据眼轮廓的电压电平。训练逻辑的一些实施方案因此可传输一连串读取/写入命令以便读取和写入去往和来自DRAM的训练序列。可使用不同的时序延迟和不同的电压偏移执行所述一连串读取/写入命令以便确定数据眼轮廓中的电压电平。举例来说,响应于来自控制器215的信令,ACSM 220可发出命令以便使用初始时序延迟将由PPGC 225产生的一个或多个序列写入到DRAM 210中的一个或多个地址。ACSM220随后可将一连串循环读取命令发出到DRAM 210中的地址。可与将不同的电压偏移值提供给接收器255或驱动器260、265同时地发出读取/写入命令。DTSM 240随后可将所循环的读取命令中的每一者的所接收的序列与所提供的训练序列进行比较以便针对初始时序延迟确定数据眼轮廓的左边缘与右边缘之间的电压电平。可改变(例如,递增或递减)所述时序延迟且可重复确定电压电平的过程。可迭代此过程以便确定时序延迟和电压电平范围上的二维数据眼轮廓。一些实施方案可替代地迭代地选择电压电平且循环经过选定电压电平的时序延迟来确定二维数据眼轮廓。

[0036] DTSM 240指令时序/电压控制逻辑280调谐接收器255或驱动器260、265的时序延迟和电压偏移,以便对应于数据眼轮廓中的提供最佳电压电平和时序延迟的位置。可基于取样训练数据中的正确样本和不正确样本的数目来确定对时序延迟或电压偏移的调节。DTSM240的一些实施方案可基于取样训练数据中的正确样本与不正确样本的预定比率来确定最佳的时序延迟和电压偏移。举例来说,DTSM 240可调谐时序延迟和电压偏移,直到由存储器PHY 205接收的正确样本与不正确样本的数目的比率处于或低于预定比率为止。DTSM 240的一些实施方案可在可确定更好的最佳训练位置的预期下使用所述预定比率来更改数据眼轮廓的形状。举例来说,可基于预定比率来扩大或收缩2D眼轮廓。对2-D数据眼轮廓的形状的其它更改也是可能的。

[0037] 本文所描述的技术的实施方案与在系统BIOS中实施且必须在数据管道上传达给存储器PHY的常规训练算法相比具有若干优势。集成的训练逻辑的一些实施方案可减少总训练时间,或允许在实施越来越复杂的训练序列的同时维持基本上恒定的训练时间。将训练逻辑集成到存储器PHY中允许将高速命令/数据产生和控制定位成靠近存储器PHY数据管道,这允许对训练序列的细粒度控制等。训练逻辑的实施方案还可通过强制固件算法开发与硬件机构设计之间的更紧密耦合来增强无种训练。还可针对不同的存储器PHY例子并行地执行训练。当将训练逻辑集成到存储器PHY中时,硅系统调试和硬件初启过程可更容易实施。

[0038] 当将训练逻辑集成到存储器PHY中时,还可改进设计过程。举例来说,可与SOC的剩余部分独立地开发并测试用于存储器PHY的训练硬件和训练算法。可在SOC中的存储器PHY与其它逻辑之间的边界处含有系统复杂性和存储器PHY行为的细节。系统配置还可支持从RTL模拟自然扩展到集成训练算法开发环境中。另外,用于存储器PHY和对应训练算法的训练硬件可封装为跨越多个SOC(包括第三方SOC)的完整IP。

[0039] 在一些实施方案中,上文所描述的设备和/或技术实施于包括一个或多个集成电路(IC)装置(还被称作集成电路封装或微芯片)(例如,上文参考图1到3所描述的存储器PHY)的系统中。可在这些IC装置的设计和制造中使用电子设计自动化(EDA)和计算机辅助设计(CAD)软件工具。这些设计工具通常表示为一个或多个软件程序。所述一个或多个软件程序包括可由计算机系统执行以进行以下操作的代码:操纵计算机系统对表示一个或多个IC装置的电路的代码进行操作,以便执行用于设计或调适制造系统来制造电路的过程的至少一部分。此代码可包括指令、数据或指令与数据的组合。表示设计工具或制造工具的软件指令通常存储在计算系统可存取的计算机可读存储媒体中。同样,表示IC装置的设计或制造的一个或多个阶段的代码可存储在同一计算机可读存储媒体或不同计算机可读存储媒体中且从所述计算机可读存储媒体进行存取。

[0040] 计算机可读存储媒体可包括可由计算机系统在使用期间存取以便将指令和/或数据提供给计算机系统的任何存储媒体,或存储媒体的组合。此类存储媒体可包括(但不限于)光学媒体(例如,压缩光盘(CD)、数字多功能光盘(DVD)、蓝光光盘)、磁性媒体(例如,软盘、磁带或磁性硬盘驱动器)、易失性存储器(例如,随机存取存储器(RAM)或高速缓冲存储器)、非易失性存储器(例如,只读存储器(ROM)或快闪存储器),或基于微机电系统(MEMS)的存储媒体。计算机可读存储媒体可嵌入计算系统(例如,系统RAM或ROM)中、固定地附接到计算系统(例如,磁性硬盘驱动器)、可移除地附接到计算系统(例如,光盘或基于通用串行总线(USB)的快闪存储器),或经由有线或无线网络耦合到计算机系统(例如,网络可存取存储装置(NAS))。

[0041] 图4是示出用于设计和制造实施根据一些实施方案的一个或多个方面的IC装置的示例性方法400的流程图。如上文所述,针对以下过程中的每一者所产生的代码存储在或以其它方式体现在非暂时性计算机可读存储媒体中以供对应的设计工具或制造工具存取和使用。

[0042] 在框402处,产生IC装置的功能规范。可通过多种编程语言或建模语言(包括C、C++、SystemC、Simulink或MATLAB)中的任一者来表示所述功能规范(常常被称作微型架构规范(MAS))。

[0043] 在框404处,使用所述功能规范来产生表示IC装置的硬件的硬件描述代码。在一些实施方案中,使用至少一种硬件描述语言(HDL)来表示所述硬件描述代码,所述硬件描述语言包括多种计算机语言、规范语言,或用于IC装置的电路的规范化描述和设计的建模语言中的任一者。所产生的HDL代码通常表示IC装置的电路的操作、电路的设计和/或组织,以及用于通过模拟来验证IC装置的正确操作的测试。HDL的实例包括模拟HDL(AHDL)、Verilog HDL、SystemVerilog HDL和VHDL。对于实施同步数字电路的IC装置,硬件描述符代码可包括寄存器传递级(RTL)代码来提供对同步数字电路的操作的抽象表示。对于其它类型的电路,硬件描述符代码可包括行为级代码来提供电路操作的抽象表示。由硬件描述代码表示的HDL模型通常经受一轮或多轮的模拟和调试来通过设计验证。

[0044] 在验证由硬件描述代码表示的设计之后,在框406处,使用合成工具来合成硬件描述代码以便产生表示或界定IC装置的电路的初始物理实现方式的代码。在一些实施方案中,所述合成工具产生一个或多个网表,其包括电路装置例子(例如,门、晶体管、电阻器、电容器、电感器、二极管等)以及电路装置例子之间的网或连接。替代地,可在不使用合成工具

的情况下手动地产生网表的全部或一部分。与硬件描述代码一样,网表可经受一个或多个测试和验证过程,之后产生最终的一组一个或多个网表。

[0045] 替代地,可使用示意性编辑器工具起草IC装置的电路的示意图,且随后可使用示意性捕获工具捕获所得的电路图,且产生表示电路图的组件和连接性的一个或多个网表(存储在计算机可读媒体上)。所捕获的电路图随后可经受一轮或多轮模拟来进行测试和验证。

[0046] 在框408处,一个或多个EDA工具使用在框406处产生的网表来产生表示IC装置的电路的存储器物理布局的代码。此过程可包括(例如)放置工具,其使用网表来确定或固定IC装置的电路的每个元件的位置。此外,布线工具依赖放置过程而添加并布线根据网表连接电路元件所需的电线。所得的代码表示IC装置的三维模型。可以数据库文件格式(例如,图形数据库系统II(GDSII)格式)表示所述代码。呈此格式的数据通常表示几何形状、文本标记,和关于呈分层形式的电路布局的其它信息。

[0047] 在框410处,将存储器物理布局代码(例如,GDSII代码)提供给制造设施,所述制造设施使用存储器物理布局代码来配置或以其它方式调适制造设施的制造工具(例如,通过掩模工具)来制造IC装置。也就是说,可将存储器物理布局代码编程到一个或多个计算机系统中,所述一个或多个计算机系统随后可完全地或部分地控制制造设施的工具的操作,或其中执行的制造操作。

[0048] 在一些实施方案中,上文所描述的技术的某些方面可由执行软件的处理系统的一个或多个处理器实施。所述软件包括存储在或另外有形地体现在非暂时性计算机可读存储媒体上的一组或多组可执行命令。所述软件可包括命令和某些数据,其当由一个或多个处理器执行时会操纵所述一个或多个处理器执行上文所描述的技术的一个或多个方面。所述非暂时性计算机可读存储媒体可包括(例如)磁盘或光盘存储装置、固态存储装置,例如快闪存储器、高速缓冲存储器、随机存取存储器(RAM),或其它非易失性存储器装置,和类似者。存储在非暂时性计算机可读存储媒体上的可执行命令可呈由一个或多个处理器解译或可以其它方式执行的源代码、汇编语言代码、目标代码,或其它命令格式。

[0049] 应注意,不需要上文在一般描述中所描述的所有活动或要素、可能不需要特定活动或装置的一部分,且可执行一个或多个其它活动,或包括除了所描述的要素之外的要素。此外,列举活动的次序不一定是执行活动的次序。而且,已经参考特定实施方案描述了概念。然而,本领域技术人员应了解,可在不脱离所附权利要求书中所陈述的本公开的范围的情况下作出各种修改和改变。因此,可在说明性意义而不是限制性意义上对待说明书和图,且希望所有此类修改包括在本公开的范围之内。

[0050] 上文已关于特定实施方案描述了益处、其它优势和问题的解决方案。然而,这些益处、优势、问题的解决方案以及可导致任何益处、优势或解决方案发生或变得更加突出的任何特征不应被解释为任何或所有权利要求的关键、所需或实质特征。另外,上文公开的特定实施方案仅是说明性的,因为所公开的标的可以按受益于本文教示的益处的本领域技术人员显而易见的不同但等效的方式加以修改和实践。除了所附权利要求书中所描述的内容之外,不希望对本文示出的构造或设计的细节进行限制。因此显然的是,可更改或修改上文所公开的特定实施方案,且所有此类变化都被视为在所公开的标的的范围内。因此,本文寻求的保护在所附权利要求书中予以陈述。

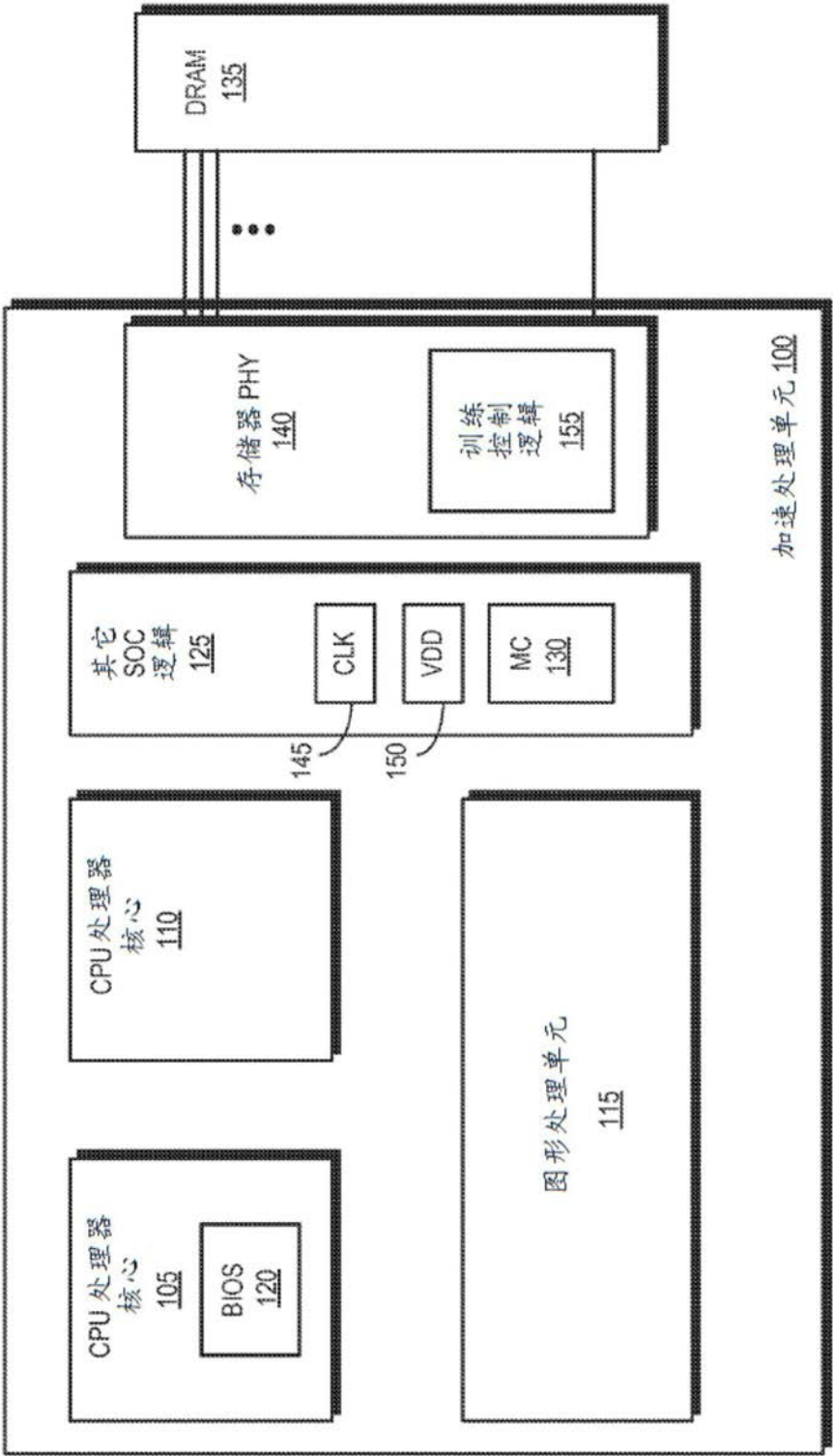


图1

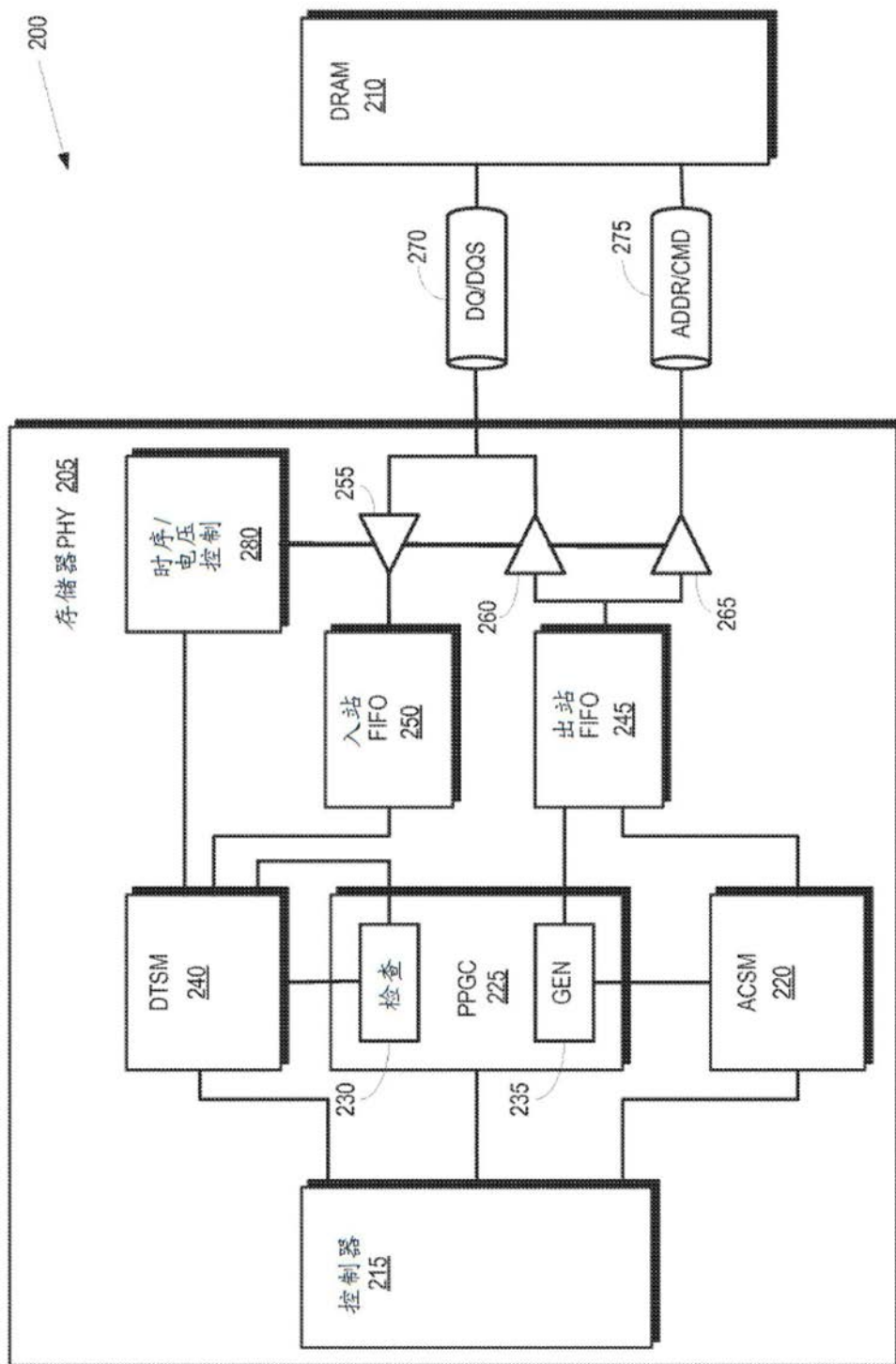


图2

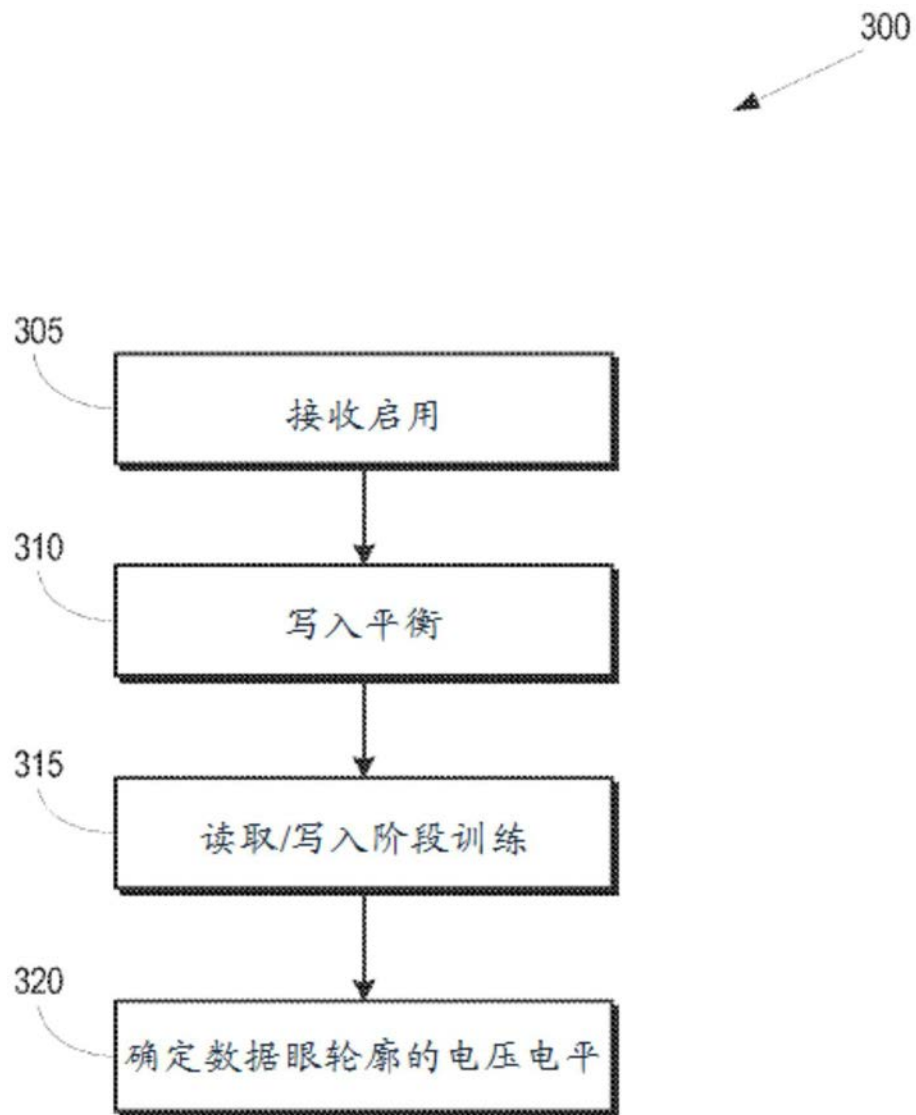


图3

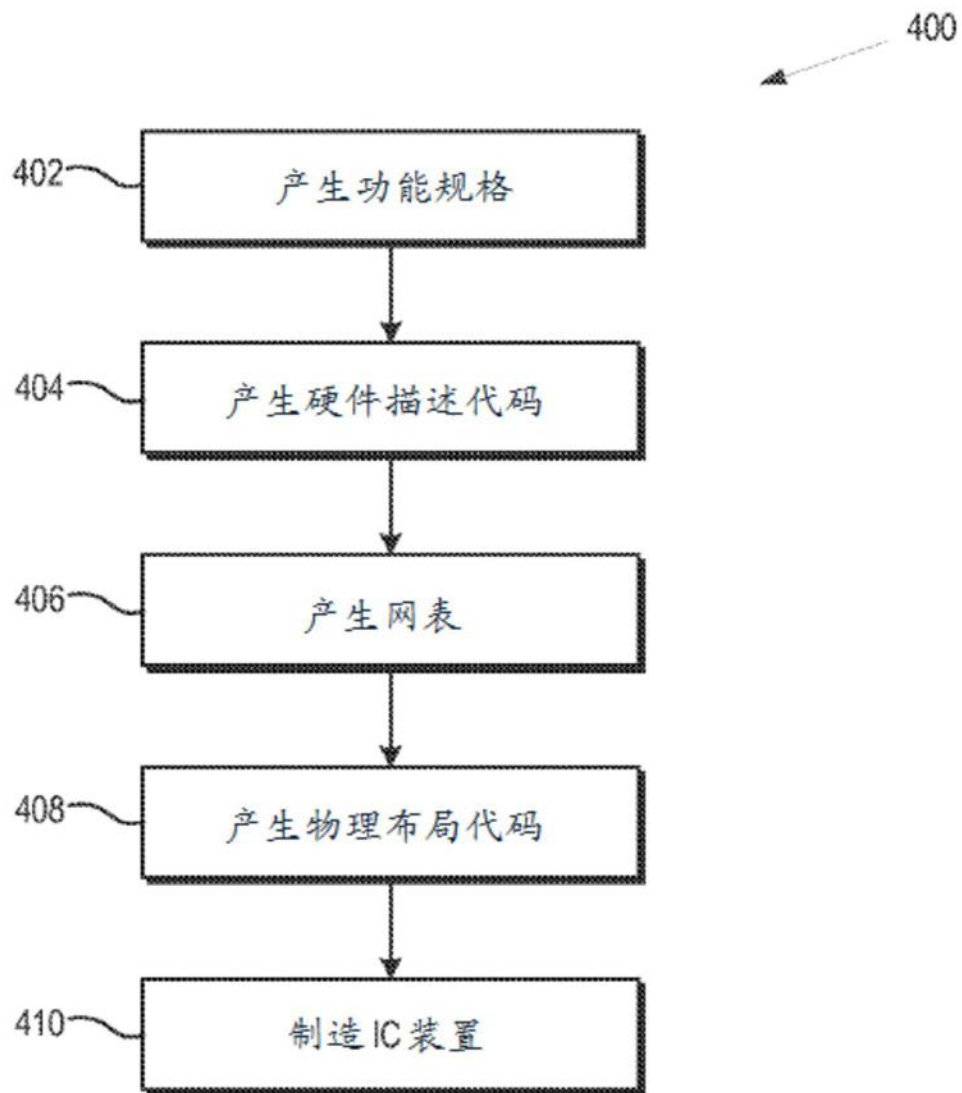


图4