



(12) 发明专利申请

(10) 申请公布号 CN 101917189 A

(43) 申请公布日 2010.12.15

(21) 申请号 201010267833.6

(22) 申请日 2006.02.03

(30) 优先权数据

11/050,644 2005.02.03 US

(62) 分案原申请数据

200680003802.1 2006.02.03

(71) 申请人 睦塞德技术公司

地址 加拿大安大略

(72)发明人 迪特尔·黑勒 托尼·迈尔

彼得·弗拉先科

(74) 专利代理机构 中科专利商标代理有限责任

公司 11021

代理人 王波波

(51) Int. Cl.

H03L 7/08 (2006.01)

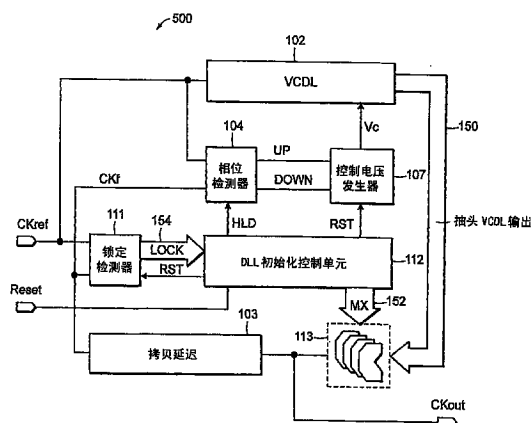
权利要求书 3 页 说明书 15 页 附图 12 页

(54) 发明名称

用于初始化延迟锁定环的方法和装置

(57) 摘要

一种延迟锁定环,包括初始化电路,该电路确保将 DLL 初始化为不会过于接近延迟与控制电压关系特性曲线的任一端的工作点。该初始化电路迫使 DLL 总是最初从初始延迟开始搜索锁定点,在一个方向上改变延迟,并迫使 DLL 跳过第一锁定点。初始化电路仅允许 DLL 从初始延迟开始在一个方向上改变压控延迟环的延迟,直到达到工作点。



1. 一种延迟锁定环,用于向参考时钟信号提供延迟,该延迟锁定环具有至少一个潜在锁定点,且该延迟锁定环包括:

锁定点临近度检测器,检测与所述至少一个潜在锁定点的临近度;

初始化控制电路,耦合至所述锁定点临近度检测器的输出端;

反相电路,包括多路复用器以及至少一个反相器,所述至少一个反相器的输出耦合至所述多路复用器输入端,所述反相电路用于使得能够根据所述初始化控制电路的输出,选择两个时钟信号中的单一时钟信号,以改变参考时钟信号的延迟;以及

可变延迟线,其输出端耦合至所述反相电路。

2. 根据权利要求1所述的延迟锁定环,其中,所述至少一个潜在锁定点是多个延迟处的多个潜在锁定点。

3. 根据权利要求2所述的延迟锁定环,其中,所述多个潜在锁定点多于两个,或者所述多个潜在锁定点在约3到5的范围中。

4. 根据权利要求1所述的延迟锁定环,其中,所述可变延迟线是压控延迟线。

5. 根据权利要求4所述的延迟锁定环,其中,所述压控延迟线包括串联连接的多个单端缓冲器。

6. 根据权利要求4所述的延迟锁定环,其中,所述压控延迟线包括串联连接的多个差分缓冲器。

7. 根据权利要求1所述的延迟锁定环,其中,所述反相电路具有单端输出。

8. 根据权利要求1所述的延迟锁定环,其中,所述反相电路具有差分输入和差分输出。

9. 根据权利要求1所述的延迟锁定环,其中,所述至少一个反相器包括多个反相器。

10. 根据权利要求1所述的延迟锁定环,还包括拷贝延迟电路,该拷贝延迟电路的输入端被配置为接收两个时钟信号中的所述单一时钟信号。

11. 根据权利要求1所述的延迟锁定环,其中,所述初始化控制电路在上电后工作。

12. 根据权利要求1所述的延迟锁定环,其中,所述初始化控制在复位后工作。

13. 一种延迟锁定环,用于向时钟信号提供延迟,该延迟锁定环具有多个延迟处的多个潜在锁定点,该延迟锁定环包括:

锁定点临近度检测器,检测与锁定点的临近度;

初始化控制电路,耦合至所述锁定点临近度检测器的输出端;以及

压控延迟线,包括串联连接的多个单端缓冲器以提供延迟线,每一单端缓冲器具有单端输入和单端输出,其中至少一个单端缓冲器连接至控制电压。

14. 根据权利要求13所述的延迟锁定环,其中,所述初始化控制电路在复位后工作。

15. 一种延迟锁定环DLL,包括:

压控延迟线;

控制电压发生器,向所述压控延迟线提供DLL控制电压;

相位检测器,按照使所述控制电压更为接近目标电压电平的方式,来进行所述控制电压的调整;以及

锁定点临近度检测器,接收第一时钟信号和第二时钟信号,该锁定点临近度检测器根据第一和第二时钟信号之间的相位差,提供锁定检测信号,该锁定点临近度检测器包括至少两个触发器和多个延迟线。

16. 一种用于初始化延迟锁定环的方法,包括:

从初始延迟开始,沿一个方向改变时钟信号的延迟,该时钟信号具有多个潜在锁定
点;

通过接收第一时钟信号和第二时钟信号并根据第一和第二时钟信号之间的相位差提
供锁定检测信号,来跳过所述多个锁定点中的第一潜在锁定点;以及

继续改变延迟以搜索工作点。

17. 一种用于初始化延迟锁定环的方法,包括:

从初始延迟开始,沿一个方向改变时钟信号的延迟,该时钟信号具有多个潜在锁定
点;

通过将控制电压传递经过多个单端缓冲器,来跳过所述多个锁定点中的第一潜在锁定
点,并继续沿同一方向改变时钟信号的延迟。

18. 一种搜索锁定点的方法,包括:

从延迟对控制电压特性的预定点开始,改变时钟信号的延迟,该延迟对控制电压特性
的预定点与延迟对控制电压特性上的最大延迟相距一定量;以及

仅增加延迟,以搜索锁定点。

19. 一种在延迟锁定环 DLL 的初始化过程期间对相位检测器输出的信号进行限制的方
法,输出的信号用于由控制电压发生器来积分以提供 DLL 控制电压,该方法包括:

接收保持信号,以在断言时提供 DLL 内正在进行初始化过程的指示;以及

如果所述保持信号被断言,则控制输出的信号,以按照使控制电压更接近位于稳定性
增强的 DLL 工作区域中的目标电压电平的方式,来进行控制电压的调整。

20. 根据权利要求 19 所述的方法,其中,输出的信号包括 UP 信号和 DOWN 信号。

21. 根据权利要求 20 所述的方法,其中,控制的步骤包括防止产生 UP 信号和 DOWN 信号
中选定的一个信号。

22. 根据权利要求 19 所述的方法,还包括接收初始化过程已经完成的指示的步骤,接
收所述指示的步骤在控制步骤之后进行。

23. 根据权利要求 22 所述的方法,其中,所述指示是所述保持信号变为非断言。

24. 根据权利要求 22 所述的方法,其中,所述控制电压的调整是电压增加调整,以及在
初始化过程的开始阶段, DLL 工作于与控制电压对受控延迟曲线上实质平坦的区域相对应
的工作区域。

25. 根据权利要求 22 所述的方法,其中,所述控制电压的调整是电压降低调整,以及在
初始化过程的开始阶段, DLL 工作于与控制电压对受控延迟曲线上实质陡峭的区域相对应
的工作区域。

26. 一种延迟锁定环 DLL,包括:

压控延迟线;

控制电压发生器,向所述压控延迟线提供 DLL 控制电压;以及

相位检测器,该相位检测器

i) 在正常工作时,输出至少两个信号,以便由所述控制电压发生器进行积分;

ii) 接收保持信号,该保持信号在断言时提供 DLL 内正在进行初始化过程的指示;以及

iii) 如果所述保持信号被断言,则控制输出的信号,以按照使所述控制电压更接近位

于稳定性增强的 DLL 工作区域中的目标电压电平的方式,来进行所述控制电压的调整。

27. 根据权利要求 26 所述的延迟锁定环,其中,输出的信号包括 UP 信号和 DOWN 信号。

28. 根据权利要求 27 所述的延迟锁定环,其中,所述控制输出的信号的步骤包括防止产生所述 UP 信号和所述 DOWN 信号中选定的一个信号。

29. 根据权利要求 27 所述的延迟锁定环,其中,所述相位检测器包括两对 UP 和 DOWN 输出,其中仅有一对输出耦合至所述控制电压发生器的输入。

30. 根据权利要求 29 所述的延迟锁定环,其中,所述一对输出以及另一对输出分别适于所述初始化过程期间的禁用和启用。

31. 根据权利要求 26 所述的延迟锁定环,其中,在完成所述初始化过程时,所述相位检测器接收完成指示。

32. 根据权利要求 31 所述的延迟锁定环,其中,所述指示是所述保持信号变为非断言。

33. 根据权利要求 31 所述的延迟锁定环,其中,所述控制电压的调整是电压增加调整,以及在初始化过程的开始阶段, DLL 工作于与控制电压对受控延迟曲线上实质平坦的区域相对应的工作区域。

34. 根据权利要求 31 所述的延迟锁定环,其中,所述控制电压的调整是电压降低调整,以及在初始化过程的开始阶段, DLL 工作于与控制电压对受控延迟曲线上实质陡峭的区域相对应的工作区域。

35. 根据权利要求 31 所述的延迟锁定环,还包括适于在所述初始化过程期间使用的粗对齐确定部件。

36. 根据权利要求 35 所述的延迟锁定环,其中,所述粗对齐确定部件是锁定检测器,且所述相位检测器适于提供细对齐确定。

用于初始化延迟锁定环的方法和装置

背景技术

[0001] 图 1 是现有技术的常规延迟锁定环 (DLL) 100 的框图。DLL 的主要功能是通过对齐两个时钟信号的上升沿而对这两个时钟信号进行同步。外部提供的时钟信号 CK 由时钟缓冲器 101 进行缓冲, 以提供基准时钟信号 CKref, 该基准时钟信号 CKref 耦合至压控延迟线 (VCDL) 102 和相位检测器 (PD) 104。压控延迟线 102 产生 DLL 输出时钟信号 CKout, 该输出信号是 CKref 的延迟形式, 并通过被称作时钟树的缓冲结构而路由至设备内的各个电路。

[0002] 反馈时钟信号 CKf 在时钟树分支的终端节点处被抽头引出, 或通过把输出时钟信号 CKout 施加到时钟树分支的拷贝即拷贝延迟电路 103 而获得, 并被反馈至 PD 104。拷贝延迟电路 103 还被称作延迟模型或时钟树分支拷贝, 再现了由时钟树的多级缓冲结构所添加到输出时钟信号 CKout 的所有延迟。该延迟包括经过逻辑门和缓冲器的所有传播延迟, 以及由长引线的寄生阻抗而引起的延迟。最终同步的反馈时钟信号 CKf 在时钟树的每个分支的末端输出。由 VCDL 102 产生的延迟是可变的, 而且可以通过施加到 VCDL 102 的可变控制电压 V_c 来控制。DLL 使用对由 VCDL 102 产生的延迟的改变能力, 通过把时钟信号 (CKref, CKf) 的上升沿对齐, 对基准时钟信号 CKref 和反馈时钟信号 CKf 进行同步。

[0003] 相位检测器 104 典型地在 UP 和 DOWN 输出信号上产生可变宽度的脉冲, 取决于基准时钟信号 CKref 和反馈时钟信号 CKf 之间的相位差。UP 和 DOWN 输出信号上的可变宽度脉冲由电荷泵 105 和耦合至电荷泵 105 的输出端的环路滤波器 106 进行积分, 以便为 VCDL 102 提供可变的控制电压 V_c 。控制电压 V_c 确定了由 VCDL 102 添加到基准时钟信号 CKref 的延迟, 从而对反馈时钟信号 CKf 和基准时钟信号 CKref 的上升沿进行对齐。同时, 电荷泵 105 和环路滤波器 106 组成了控制电压发生器 107。

[0004] 图 2 是示出了典型控制电压 V_c 与受控延迟之间的特性曲线的图表。该特性曲线是非线性的, 并且包括平坦区 202、最优区 200 和陡峭区 204。在平坦区 202 中, 相对小的延迟范围需要控制电压 V_c 发生宽的变化。

[0005] 在陡峭区 204 中, 控制电压 V_c 的小变化会提供大的延迟范围。因此, VCDL 在陡峭区 204 中具有很高的灵敏度, 因为控制电压 V_c 上的小噪声干扰会导致延迟发生大的变化, 从而导致时钟抖动的增加。在这种高灵敏度的情况下, 更加难以提供稳定、非振荡的环路操作。

[0006] 在“最优区”200 中, 延迟的变化相对于控制电压的变化是适中的。因此, 工作在“最优区”中的 DLL 100 不会具有振荡、漂移或累积噪声。

[0007] 锁定点是该特性曲线中能够锁定 DLL 的任意点。该特性曲线上可以存在多个锁定点。工作点是 DLL 在正常操作期间被锁定的锁定点。设计 DLL 时一个重要的方面是, 在该特性曲线中选择正确的工作点, 并在上电或复位后快速地把 DLL 引入且锁定至该工作点。这个过程典型地被称作 DLL 初始化。适当的 DLL 初始化确保了良好的 DLL 性能和稳定的锁定。

[0008] 选择正确的工作点把控制电压 V_c 设置为与稳定工作区相关的目标电压电平。为了确保稳定的 DLL 操作, 应当把 DLL 初始化为 VCDL 延迟与控制电压特性曲线的“最优

区” 200 中的工作点。

[0009] 在 DLL 已经达到该工作点后,由于工作条件的变化,例如温度和电源的变化,该工作点会发生移动。因此, DLL 设计中另一个重要的方面是,在工作条件发生变化时,保持工作点处于延迟与电压特性曲线上的锁定点的预定界限内。控制电压 V_c 的变化被限制为最多是电源电压的变化,通常控制电压 V_c 的变化小于电源电压的变化。因此,图 2 所示的延迟与控制电压特性曲线在两端是不是无限的,而且当工作条件变化时,工作点可能会漂移至该特性曲线的左界限或右界限,而且 DLL 将最终失去锁定。如果把 DLL 初始化为过于接近该特性曲线两端中任一端的工作点,则上述情况出现的概率尤其高。

[0010] 优选地把工作点锁定至该特性曲线左侧的锁定点,因为这是具有更好抗噪性的区域。然而,如果工作点过于接近该特性曲线的左端,则 DLL 可能由于工作条件的变化而达到该特性曲线的左界限。这种情况在图 3A 和 3B 中示出。

[0011] 图 3A 是示出了锁定点 300 的图表,该锁定点 300 接近控制电压与延迟关系特性曲线的左端。图 3B 是与图 3A 中的控制电压与延迟关系特性曲线相对应的时钟信号时序图。工作点在额定条件下处于锁定点 300 处。需要 VCDL 的延迟范围 302 对工作条件的变化进行补偿。参考时钟信号时序图,反馈时钟信号 CKf 304 的上升沿漂移与图 3A 的图表所示的延迟范围 302 相对应。延迟范围 302 中的最小延迟 306 超出了 VCDL 所产生的全部延迟范围 310,如延迟范围中的间隙 308 所示。因此,工作点可以移动至 VCDL 范围的末端,从而导致 DLL 失去锁定。

[0012] 另一个潜在的风险是, DLL 在启动期间,在不具有搜索方向限制的情况下从该特性曲线上的随机点开始搜索锁定点。由于外部提供的时钟信号 CK 是自由运行的,所以反馈时钟信号 CKf 与基准时钟信号 CKref 之间的初始相位关系在复位或上电后是未知的。此外,在上电或复位后,初始 DLL 未锁定工作点的位置是未知的,而且可以是该特性曲线上的任意处。因此,取决于 PD 104(图 1)首先检测到反馈时钟信号 CKf 的上升沿还是基准时钟信号 CKref 的上升沿,最初可以增加或减小 VCDL 延迟。因此,对 VCDL 进行最初调整的方向是不可预测的。

[0013] 图 4A 是示出了在某个搜索方向上对锁定点进行最初搜索的图表,该搜索导致在达到锁定前碰到 VCDL 102(图 1)的延迟界限。图 4B 是与图 4A 所示搜索相对应的时钟信号时序图。对锁定点的搜索从随机搜索点 400 开始。如果 DLL 从接近该特性曲线末端的点启动并朝着该末端前进,则在达到锁定前可能会碰到 VCDL 的延迟界限。在图 4A-4B 所示的示例中,在初始化期间, DLL 不可预测地向最近的锁定点 402 移动,而该锁定点 402 在 VCDL 之外且不能达到。例如,这种情形将会在如下情况时发生:相位检测器 104(图 1)最初产生的 UP/DOWN 脉冲使 DLL 100 在最近的锁定点 402 的方向上移动,该锁定点 402 在 VCDL 范围 310 之外。

[0014] DLL 中可变的 VCDL 延迟范围也是重要的。通常,计算可变 VCDL 延迟范围,从而最小延迟与稍高于 DLL 规范所需的时钟频率相对应,而最大延迟与稍低的时钟频率相对应。计算可变 VCDL 延迟,以确保富余 (margin)。对于被设计用于在宽的时钟频率范围上工作的 DLL 来说,即当时钟周期不是常数值且所有可能值应该被相同的 VCDL 所容纳时,该 VCDL 必须产生甚至更宽的延迟范围。结果,对于具有具体频率的时钟信号, VCDL 特性曲线上典型地存在多个可能的锁定点。对于更高的时钟频率, VCDL 可以产生大于时钟周期倍数的延

迟。目标是锁定至可确保稳定的锁定条件及较低的输出时钟抖动的点。在多数情况下,选择 VCDL 的延迟范围,使得 DLL 特性曲线上可能的锁定点的数目大于 2,但不超过大约 3 至 5。如果存在过多的锁定点,则这些锁定点将会在该特性曲线上靠近地共存,而且如果受到噪声干扰,DLL 会开始从一个锁定点跳至另一个锁定点,从而暂时失去锁定。

发明内容

[0015] 在已知技术中,用于确保 DLL 的稳定锁定和良好性能的解决方案包括复杂的结构,例如双环结构。还使用多相形式的时钟信号,或最简单情况下的时钟反相。然而,使用它们是为了使 VCDL 中的相位抽头的数目最小化和 / 或提供具有更少级数的延迟线。相位抽头用于增大流水线级同步的灵活性,但不是为了实现 DLL 自身的稳定锁定和稳定性能。

[0016] 典型地,DLL 的设计者不愿花费过多的时间来处理这个初始化的“次要”问题,而且他们相信传统的“已证实”的方法。因此,期望提供一种 DLL 初始化方法,能够减轻传统方法的问题。

[0017] 提供了一种用于确保将 DLL 初始化为正确的工作点的方法和装置,该工作点不会过于接近延迟与控制电压关系特性曲线的任一端。初始化电路迫使 DLL 总是从与延迟和电压关系特性曲线的一端相对应的初始延迟开始搜索锁定点,并允许 DLL 仅在一个方向上改变受控的延迟,直到达到最终锁定。在达到最终锁定后,DLL 可以根据需要而增大或减小受控延迟,从而动态地维持锁定点。根据本发明提供的方法,还迫使 DLL 跳过第一锁定点,而且通过对 DLL 内部时钟信号进行移相而使该跳过过程更容易和更快。

[0018] 在本发明的特定实施例中,DLL 从延迟与控制电压关系特性曲线上与最小延迟相对应的端点开始搜索锁定点。在初始化期间,锁定点的搜索方向被限制为仅增大延迟,即迫使相位检测器仅产生与增大延迟相对应的 UP 和 DOWN 信号的组合,这取决于相位检测器的内部结构。DLL 增大延迟,直到 DLL 达到第一锁定点。然后,对延迟的时钟信号进行移相。在一个实施例中,将时钟信号反相。这迫使 DLL 跳过第一锁定点,并前进至下一个锁定点。作为移相的结果,相位检测器检测到其两个输入时钟信号之间的新的相位差,并开始再次产生 UP 和 DOW 信号,直到达到特性曲线上的工作点。这导致 VCDL 延迟时间的总增大大约等于一个完整时钟周期的持续时间。在达到工作点后,可以增大或减小延迟变化。在 DLL 初始化后,DLL 对输入时钟信号相位漂移和工作条件变化进行补偿,从而动态地维持稳定的锁定。

[0019] 在其他实施例中,DLL 从延迟与控制电压关系特性曲线中接近最大延迟的点开始搜索锁定点。在初始化期间,锁定点的搜索方向被限制为仅减小延迟,即迫使相位检测器仅产生与减小延迟相对应的 UP 和 DOWN 信号的组合,这取决于相位检测器的内部结构。这个实施例应用于如下情况:延迟与控制电压关系特性曲线的非线性不如图 2 的示例中的非线性,并具有与边界控制电压值相对应的有限最大延迟。换句话说,应用于 VCDL 能够在最小和最大延迟上良好工作时,即在两个控制电压边界值上良好工作时。在这种情况下,也可以应用 VCDL 输出时钟反相和移相。

[0020] 延迟锁定环包括锁定检测器和与所述锁定检测器的输出端相连的初始化控制。所述锁定检测器检测与锁定点的临近度。所述初始化控制从初始延迟开始在一个方向上改变延迟。所述初始化控制跳过第一锁定点,并在检测到与所述第一锁定点临近时继续在所述

同一个方向上改变所述延迟,以搜索工作点。在检测到与所述工作点临近时,所述初始化控制启用对所述延迟的增大和减小。所述工作点可以是第二锁定点。

[0021] 所述锁定检测器可以包括具有不同预置时间间隔的多个级,每一级指示具有不同精确度的锁定点临近度。初始化状态可以是上电或复位。通过对时钟信号进行移相(在最简单的情况下是对信号进行反相),跳过所述第一锁定点。时钟信号的相位的移动量是时钟信号周期的片段。压控延迟线抽头输出用于对时钟信号进行移相。可以通过对所述延迟锁定环中的内部时钟信号执行移相,对所述时钟信号进行移相。所述内部时钟信号可以是压控延迟线输入时钟信号或压控延迟线输出时钟信号。

[0022] 根据基准时钟和延迟的反馈时钟的沿对齐来检测锁定点临近度。通过施加多个预置时间间隔中至少一个来分析所述临近度。所述预置时间间隔的值可以基于压控延迟线中的级的拷贝,而且可以小于时钟周期的四分之一或两个相邻压控延迟线抽头输出之间的延迟时间的片段。

附图说明

[0023] 根据下文对附图中所示的本发明的优选实施例进行更为具体的描述,本发明的前述和其他目标、特性和优点将会变得明显,附图中相似的附图标记表示相同的部分。附图不需按比例绘制,而把重点放在对本发明原理解释。

[0024] 图 1 是现有技术的常规延迟锁定环(DLL)的框图;

[0025] 图 2 是示出了典型的控制电压 V_c 与 VCDL 延迟之间的关系特性曲线的图表;

[0026] 图 3A 是示出了接近控制电压与延迟关系特性曲线的左端的锁定点的图表;

[0027] 图 3B 是与图 3A 中的控制电压与延迟关系特性曲线相对应的时钟信号图;

[0028] 图 4A 是示出了在某个搜索方向上对锁定点进行搜索的图表,该搜索导致在达到锁定前碰到 VCDL 的延迟界限;

[0029] 图 4B 是与图 4A 所示锁定点搜索相对应的时钟信号图;

[0030] 图 5 是根据本发明原理的延迟锁定环(DLL)的实施例的框图,包括用于对 DLL 进行初始化的 DLL 初始化控制;

[0031] 图 6A 是示出了控制电压与延迟关系特性曲线上的初始化过程的图表,其中使用了图 5 所示的 DLL 中的 DLL 初始化控制;

[0032] 图 6B 是与图 6A 所示初始化过程相对应的时钟信号图;

[0033] 图 7 是图 5 所示 VCDL 的单端实施例的框图;

[0034] 图 8 是差分端 VCDL 的框图;

[0035] 图 9 是锁定检测器的实施例的示意图;

[0036] 图 10 是示出了图 9 所示锁定检测器的一级中的信号的时序图;

[0037] 图 11 是示出了用于对 DLL 进行初始化的算法的流程图;以及

[0038] 图 12-15 是根据本发明原理的包括 DLL 初始化的 DLL 的备选实施例的框图。

具体实施方式

[0039] 下文描述本发明的优选实施例。

[0040] 在下面的描述中,提出多个具体细节以提供对本发明的完整理解。然而,可以理解

的是本发明可以在不具有这些具体细节的情况下得以实践。在其他实例中,没有详细描述或示出公知结构和/或过程,以便使本发明清晰。在描述和附图中,相似的附图标记表示相似的结构或过程。一般地,延迟锁定环(DLL)的操作是本领域中公知的,所以不会做进一步的描述,除非需要对本发明的方面进行阐述。

[0041] 图5是根据本发明原理的延迟锁定环(DLL)500的实施例的框图,包括用于在上电或复位后对DLL 500进行初始化的DLL初始化控制112。如同结合图1所示现有技术的DLL所描述的那样,DLL 500包括压控延迟线(VCDL)102、相位检测器(PD)104、控制电压发生器107、以及拷贝延迟103。DLL还包括多路复用器113、锁定检测器111以及用于在上电或复位后对DLL 500进行初始化的DLL初始化控制112。

[0042] 把基准时钟信号CKref耦合至压控延迟线(VCDL)102的输入端、相位检测器(PD)104的两个输入端之一、以及锁定检测器111。VCDL 102产生抽头相移输出时钟信号116,每一个信号116都是基准时钟信号CKref的延迟形式。取决于从DLL初始化控制112输出的MX代码152,通过多路复用器113从抽头VCDL输出116中选择DLL输出时钟信号CKout。例如,通过被称作时钟树的缓冲结构把输出时钟信号CKout路由至半导体芯片上的各个电路。

[0043] 反馈时钟信号CKf是基准时钟信号CKref被VCDL 102和拷贝延迟电路103延迟后的信号。反馈时钟信号CKf在时钟树分支的终端节点上抽头。备选地,可以通过把输出时钟信号CKout施加到时钟树分支的拷贝,这里是指拷贝延迟电路103,而获得反馈时钟信号CKf。反馈时钟信号CKf被反馈至PD 104和锁定检测器111。拷贝延迟电路103还被称作延迟模型或时钟树分支拷贝,再现由时钟树的多级缓冲结构添加到CKout信号的所有延迟。该延迟包括经过逻辑门和缓冲器的所有传播延迟,以及由长引线的寄生阻抗而引起的延迟。反馈时钟信号CKf复制了时钟树分支的终端节点处的时钟信号。例如,在半导体芯片中,终端节点处的时钟信号被施加到同步模块的时钟输入端,而且这些时钟信号通常与基准时钟信号CKref同步。

[0044] 把DLL输入或基准时钟信号CKref耦合至VCDL 102。VCDL 102包括具有可变延迟的类似缓冲器链。该链可以包含几十个或甚至几百个缓冲器。VCDL 102所产生的延迟是可变的,而且可以通过可变控制电压Vc来控制,其中可变控制电压Vc通过控制电压发生器107而施加到VCDL102中的每一个缓冲器。

[0045] 在初始化过程开始时,把控制电压Vc设置为边界电压电平(端点),即设置为VCDL 102的延迟被设置为最小延迟处的电压电平。例如,该边界电压电平可以是电源电压或地。把DLL初始化控制112所产生的RST信号耦合至控制电压发生器107以设置初始控制电压Vc。例如,可以通过单晶体管开关把Vc节点与电源节点(干线)或地短路而执行对初始控制电压Vc的设置。

[0046] 相位检测器104产生UP和DOWN输出信号上的可变宽度脉冲,取决于基准时钟信号CKref与反馈时钟信号CKf之间的相位差。把这两个时钟信号耦合至相位检测器104的输入端。来自相位检测器104的UP和DOWN输出信号上的可变宽度脉冲由控制电压发生器107进行积分,以提供直流(DC)模式控制电压Vc。在一个实施例中,正如结合图1中的DLL100而描述的那样,控制电压发生器107可以包括电荷泵105和环路滤波器106。控制电压发生器107通过应用本领域技术人员所公知的滤波和电压电平移位操作,对PD输出信

号 (UP、DOWN) 进行积分。稍后结合图 7 和 8 来描述 VCDL 的实施例。

[0047] 待审美国专利申请“High Output Impedance Charge Pump for PLL/DLL”, Diet er Haerl e(美国专利申请 No.60/528,958) 中描述了控制电压发生器中的电荷泵的实施例,将其全部内容在此引入作为参考。电荷泵的另一个示例可以在“*A 2-1600MHz 1.2-2.5V CMOS ClockRecovery PLL with Feedback Phase-Selection and AveragingPhase-Interpolation for Jitter Reduction*”, Patrik Larsson, 1999 IEEE ISSCC, WA 20.6, 0-7803-5129-0/99, Fig. 20.6.3 中找到,将其全部内容在此引入作为参考。

[0048] 相位检测器 104 可以是产生如下输出信号 (UP、DOWN) 的任意相位检测器:该输出信号 (UP、DOWN) 在电特性(例如电压电平或脉宽)上与施加到输入端的时钟信号 (CKref、CKf) 之间的相位差成比例。相位检测器是本领域技术人员所公知的,而且不在本发明的范围之内。“*An All-Analog Multiphase Delay-Locked Loop Using a Replica Delay Line for Wide Range Operation and Low-Jitter Performance*”, Yongsam Moon et al., JSSC Vol. 35, No. 3, March 2000, pp 377-384 中描述了相位检测器的示例,将其全部内容在此引入作为参考。

[0049] DLL 100 使用对 VCDL 102 产生的延迟进行改变的能力,通过把基准时钟信号 CKref 和反馈时钟信号 CKf 各自的上升沿进行对齐,从而对这两个信号进行同步。控制电压 Vc 确定了将由 VCDL 102 添加到基准时钟信号 CKref 的延迟,以对齐反馈时钟信号 CKf 和基准时钟信号 CKref 的上升沿。

[0050] 还把基准时钟信号 CKref 和反馈时钟信号 CKf 耦合至锁定检测器 111 的输入端。锁定检测器 111 对输入时钟信号 (CKref、CKf) 的上升沿的相互定位进行估计,并产生 LOCK 指示信号 154,该信号指示所述两个时钟信号的上升沿之间的定时差。LOCK 指示信号 154 可以是单比特信号或多比特代码。稍后结合图 9 来讨论锁定检测器 111 的实施例的内部结构和操作。

[0051] 多路复用器 113 是本领域中已知的任意适合类型的多路复用器,它根据多比特代码 (MX 152) 的值选择多个输入信号(本说明书中为抽头 VCDL 输出 116)之一传输至多路复用器 113 的单一输出端(本说明书中为 Ckout)。稍后结合图 7 和 8 来描述抽头 VCDL 输出。

[0052] DLL 初始化控制 112 管理初始化过程,并负责维持初始化过程的适当顺序。DLL 初始化控制 112 还根据从锁定检测器 111 接收到的 LOCK 指示信号 154 而选择代码 (MX 152) 和信号 (HLD、RST) 的适当值。

[0053] 在上电或复位后, DLL 初始化控制 112 断言 RST 信号以复位锁定检测器 111 和控制电压发生器 107。把从控制电压发生器 107 输出的控制电压 Vc 设置为提供最小 VCDL 延迟的电压电平。DLL 初始化控制 112 还断言被耦合至相位检测器 104 的 HLD 信号。在断言 HLD 信号时,相位检测器 104 只可以通过产生适当的 UP/DOWN 信号而增加 VCDL 102 的延迟。在根据锁定指示信号 154 的状态而检测出工作点与第一锁定点接近时, DLL 初始化控制单元 112 输出适当的 MX 代码 152,以切换输出时钟信号 Ckout 的相位。在相位被切换后,相位检测器 104 通过产生适当的 UP/DOWN 信号来修改控制电压 Vc,从而继续增大 VCDL 延迟,直到达到由锁定指示信号 154 所指示的下一个锁定点为止。

[0054] 多路复用器 113 具有多个输入端和单一输出端。MX 代码 152 选择抽头 VCDL 输出

信号之一以传递至单一输出端。一次仅有一个抽头 VCDL 输出信号能够被传递至多路复用器的输出端,该信号与 MX 代码的当前值相对应。相邻抽头 VCDL 输出之间存在有限数目的延迟级,因此,多个抽头 VCDL 输出创建了“相位栅格”,其中两个相邻抽头信号之间的相位差相对较小,是时钟信号周期的片段。在相邻抽头之间的小相位差的情况下,必须跳过相对大数目的抽头,以产生更大的相移。因此,如果多路复用器的输出端所需的相移接近 180 度(大跳),则使用抽头 VCDL 信号的补码(complement),并且如果需要的话,然后使用抽头之间的小相位差(较小的跳)执行更加精确的调整。

[0055] 在达到下一锁定点后,初始化过程完成。DLL 初始化单元 112 去断言 HLD 信号,而且相位检测器 104 可以在 DLL 的正常操作期间增大或减小 VCDL 102 的延迟。

[0056] 因此,DLL 初始化控制单元 112 控制初始化过程,使得 VCDL 延迟以最小延迟开始,该延迟增大至第一锁定点,跳过第一锁定点并进一步增大延迟,直到 DLL 500 被锁定至第二锁定点。稍后结合图 11 来讨论 DLL 初始化控制单元 112 的更多操作细节。

[0057] 图 6A 是示出了控制电压与延迟关系特性曲线上的初始化过程的图表,其中使用了图 5 所示的 DLL 500。图 6B 是与图 6A 中特性曲线所示初始化过程相对应的时钟信号图。结合图 5 来描述图 6A 和 6B。

[0058] 最初把控制电压 V_c 复位至边界电压电平,即复位至与 VCDL 102 所产生的最小延迟相对应的电压。例如,边界电压电平可以是电源电压或地。

[0059] 在复位或上电后,由于控制电压 V_c 根据从 PD 104 输出的 UP/DOWN 信号而增大,所以压控延迟线(VCDL)102 从图 6A 所示的最小延迟点 602(延迟与控制电压关系特性曲线上最左边的点)开始沿方向 600 增大延迟。

[0060] 在初始化过程开始时,锁定检测器 111 对两个时钟信号(CKref、CKf)的上升沿的相对位置进行估计。根据这个估计,锁定检测器 111 产生 LOCK 信号 154,该信号指示两个时钟信号上升沿的临近度。在一个实施例中,LOCK 信号 154 是多比特代码。LOCK 信号由 DLL 初始化控制单元 112 所接收。DLL 初始化控制单元 112 根据 LOCK 信号 154 的值而产生多比特输出选择代码 MX 152。输出选择代码 MX 152 用于选择从 VCDL 102 接收到的抽头相移输出信号 116 之一。

[0061] 从最小延迟点 602 开始,在一个方向上逐渐增大通过 VCDL 102 的延迟。在初始化期间,对从 DLL 初始化控制单元 112 输出、并被耦合至 PD 104 的 HLD 信号进行断言,从而保持 PD 104 处于初始化状态。在初始化状态中,PD 仅产生 UP 或 DOWN 信号,从而仅通过适当修改控制电压 V_c 而增大 VCDL 延迟。初始化过程继续进行,直到以预置的临近度将基准时钟信号 CKref 和反馈时钟信号 CKf 的上升沿对齐。该临近度被设置为大致上小于基准时钟信号 CKref 的半个周期。

[0062] 在检测到基准时钟信号 CKref 和反馈时钟信号 CKf 的上升沿以所选的临近度而对齐后,通过适当地选择输出选择代码 MX 152, DLL 初始化控制单元 112 对输出时钟信号 CKout 进行反相(即针对 50% 占空比的时钟信号是移相 180 度)。因此,跳过了第一锁定点 604。受 DLL 初始化控制单元 112 控制的 HLD 信号继续保持 PD 104 处于初始化状态,导致继续增大 VCDL 延迟。

[0063] 在已经跳过第一锁定点 604 后, DLL 500 通过逐渐增大控制电压 V_c 而继续增大延迟,直到完成把基准时钟信号 CKref 和反馈时钟信号 CKf 的上升沿精确地对齐。该精确对齐

由 LOCK 信号 154 来指示,或由 PD 104 的 UP 和 DOWN 输出信号的状态来指示。如果 PD 104 用作前一句所述初始化过程中的精确对齐指示设备,PD 104 可以具有专门用于初始化且不耦合至控制电压发生器 107 的输入端的分离的 UP 和 DOWN 输出对。取决于 PD 104 的方案,这可能是必需的,因为与控制电压发生器 107 的输入端相连的 UP 和 DOWN 输出可能在初始化期间被禁用。

[0064] 在达到第二锁定点 606 后, HLD 信号的状态被切换至去断言,从而允许 PD 104 的正常操作。然后,允许 DLL 500 根据需要而增大或减小 VCDL 延迟。通过释放 PD 104,初始化过程终止,且 DLL 的正常操作得以启用。

[0065] 在一个实施例中,锁定指示信号 154 是多比特锁定代码。多比特 LOCK 代码 154 允许 DLL 500 通过选择与期望的锁定点 606 相对应的抽头 VCDL 输出 116 而快速地向期望的锁定点移动。这允许快速地达到第二(期望的)锁定点 606,而不需要等待控制电压 V_c 逐步改变。在备选实施例中,LOCK 信号 154 是单一比特。

[0066] 图 7 是图 5 所示 VCDL 102 的单端实施例的框图。VCDL 102 包括串联的多个单端缓冲器 115,从而提供了延迟线。每一个缓冲器 115 具有单端输入和单端输出。VCDL 102 中一个缓冲器 115 的单端输出被耦合至后续缓冲器的单端输入。控制电压 V_c 直接提供给每一个缓冲器 115,并确定由每一个缓冲器所产生的延迟时间。为了简便起见,图 7 中未示出控制电压 V_c 与每一个缓冲器 115 的连接。在这个实施例中,VCDL 输出信号 117 也是单端的。反相的(对于 50%占空比的时钟,移相 180 度)VCDL 输出信号 117'通过反相器 109 的输出端提供,而反相器 109 的输入端耦合至 VCDL 输出信号 117。

[0067] 通过缓冲器组 114 的抽头输出来获得适度延迟的基准时钟信号 CKref。多个抽头 116 组成了图 5 所示的抽头 VCDL 输出 116。每一个抽头 116 是延迟的或移相的基准时钟信号 CKref。在本发明的不同实施例中,VCDL 102 可以输出多个抽头输出 116 或单一的输出信号 117,具有或不具有其相应的补码 117'。

[0068] 图 8 是 VCDL 102 的差分端实施例的框图。在这个实施例中,VCDL102 包括受控制电压 V_c 控制的多个差分缓冲器 115a,其中控制电压 V_c 耦合至每一个差分缓冲器 115a(为了简便起见未示出)。差分缓冲器 115a 是串联的。在这个实施例中,差分缓冲器 115a 具有差分输入端和差分输出端。第一缓冲器 115b 具有用于接收单端基准时钟信号 CKref 的单端输入端和差分输出端。VCDL 102 的差分端实施例趋于获得比 VCDL 的单端实施例更好的抗噪性。此外,VCDL 输出 117a 和每个抽头输出 116a 具有信号及其补码。利用所提供的每一个抽头输出信号的补码,不需要图 7 中的单端实施例所示的反相器 109。VCDL 102 可以具有多个抽头差分输出端 116a,或具有单一的差分输出端 117a。

[0069] 图 9 是锁定检测器 111 的实施例的示意图。锁定检测器 111 提供了锁定指示信号(LC) 154,该信号基于基准时钟信号 CKref 和反馈时钟信号 CKf 之间的相位差而指示 DLL 与锁定点的接近程度。锁定检测器 111 中的每一级 118 在输入端接收两个时钟信号(CKref、CKf),并输出单比特 LOCK 信号(LC),该信号指示每一级 118 的输入时钟信号之一的上升沿与其另一输入时钟信号的上升沿之间的时间差。

[0070] 每一级 118 是自包含(self-containing)单元,它包括两个触发器 119a、119b 以及三个延迟线 122a、122b、122c。延迟线 122a、122b、122c 中的每一个具有嵌入在该延迟线中的各自的延迟时间间隔(T_1 、 T_2)。在最简单的实施方式中,锁定检测器 111 具有输出单

比特锁定指示信号的单一级 118。

[0071] 图 10 是示出了图 9 所示锁定检测器 111 的一级中的信号的时序图。图 10 结合图 9 一同使用,以解释锁定检测器 111 的操作。假定基准时钟信号 CKref 的上升沿位置和延迟的基准时钟信号 CKref_del 的上升沿位置是稳定的。该时序图示出了反馈时钟信号 CKf 的上升沿从左向右的移动,从“早”(‘E’)位置移动到“晚”(‘L’)位置。E 位置在基准时钟信号 CKref 的上升沿之前,而 L 位置在延迟的基准时钟信号 CKref_del 的上升沿之后。

[0072] 参考图 9,在每一级 118 中,级 118 的输入时钟信号 CKref 由具有延迟时间 T1 的延迟线 122c 进行延迟。延迟的输入时钟信号 CKref_del 被耦合至触发器 119a、119b 的“D”输入端。触发器 119a 的时钟输入端耦合至反馈时钟信号 CKf。触发器 119b 的时钟输入端耦合至延迟的反馈时钟信号 CKf_del,该信号已通过具有延迟时间间隔 T1 的延迟线 122a 和具有延迟时间间隔 T2 的延迟线 122b 而被延迟。延迟线 122a、122b 是串联的。

[0073] 延迟时间间隔 T1 和 T2 被嵌入延迟线 122a-c,而且可以是相同或不同的值。取决于应用,还可以修改延迟时间间隔。一般地,锁定检测器 111 包括具有与图 9 所述类似结构的多个级 118。然而,对于不同的实施方式,级 118 可以在嵌入延迟线 122a-c 的 T1 和 T2 延迟值方面不同。T1 和 T2 的优选值是基准和反馈时钟信号 (CKref、CKf) 的周期的片段,该片段小于最高时钟信号频率上的时钟周期的四分之一。在一些实施例中,该延迟时间间隔小于两个相邻 VCDL 抽头 116、116a 之间的延迟时间的一半。对于其他实施例,延迟时间间隔稍大于两个相邻 VCDL 抽头 116、116a 之间的延迟时间的一半。

[0074] 在所实施例中,延迟时间间隔 T1 和 T2 是不同的。如果期望在初始化期间反馈时钟信号 CKf 的上升沿从一侧(例如图 6 的时序图中的左侧)“逼近”基准时钟信号 CKref 的上升沿,则延迟时间间隔 T2“打开锁定窗口”且延迟时间间隔 T1 构成锁定检测器的富余。通常,T1 和 T2 是不同的值。典型地,在这个示例中,选择延迟时间间隔 T1 小于延迟时间间隔 T2,以确保用于指示锁定点正在逼近的“早警告”,并确保对锁定点另一侧更紧密的控制。

[0075] 典型地,延迟时间间隔 (T1、T2) 较短,因为需要占据大量的硅面积开销以获得更长的延迟。然而,延迟时间间隔不能过短,因为 LC 信号 154 需要在 VCDL 延迟由 DLL 500 进行调整时的多个时钟周期中保持稳定。

[0076] 参考图 9 的示意图,当反馈时钟信号 CKf 的上升沿处于“E”位置时,反馈时钟信号 CKf 的上升沿关于基准时钟信号 CKref 是早的。LC 信号 154 是‘0’,因为反相器 120 的输出端上的 ER 信号是‘1’且触发器 119b 的 Q 输出端上的 LT 信号是‘0’。

[0077] AND 逻辑门 121 把 ER 和 LT 信号进行组合,以在输出端提供 LC 信号。LT 信号上的‘0’和 ER 信号上的‘1’的组合导致 AND 逻辑门 121 的输出端处 LC 信号上的‘0’。

[0078] 当 VCDL 延迟增大时,反馈时钟信号 CKf 的上升沿向右移动并达到“i”位置。LC 信号切换至‘1’,因为‘ER’信号和‘LT’信号均为‘1’。

[0079] 参考该示意图,延迟的反馈时钟信号 CKf_del 也达到其相应的“i”位置。在延迟的反馈时钟信号 CKf_del 的上升沿处,延迟的基准时钟信号 CKref_del 是‘1’,而且触发器的 D 输入端上的‘1’被锁存至触发器 119b 的 Q 输出端。LT 信号上的‘1’和 ER 信号上的‘1’的组合导致 AND 逻辑门 121 的输出端处 LC 信号上的‘1’。

[0080] LC 信号维持为‘1’的同时,反馈时钟信号 CKf 的上升沿的位置继续向图 10 的示例中的右侧移动,直到该上升沿达到“ii”位置。在‘ii’位置,延迟的基准时钟信号 CKref_

del 是 ‘1’，因为触发器 119a 的 D 输入端是 ‘1’，反馈时钟信号 CKf 的下一个上升沿把 ‘1’ 定时至触发器 119a 的 Q 输出端，而且反相器 120 输出端上的 ER 信号切换至 ‘0’。LT 信号上的 ‘1’ 和 ER 信号上的 ‘0’ 的组合导致 AND 逻辑门 121 的输出端处 LC 信号上的 ‘1’。

[0081] LC 信号的状态提供了对反馈时钟信号 CKf 的上升沿与基准时钟信号 CKref 的上升沿之间的时间的指示。在反馈时钟信号 CKf 的上升沿与 CKref 时钟信号的上升沿的距离处于时间 T2 和 T1 内时，即在位置 (i) 和 (ii) 之间时，LC 信号维持 ‘1’。

[0082] 如果多个级 118 包括在锁定检测器 111 中，且每一级从 VCDL 102 接收相同的反馈时钟信号 CKf 和基准时钟信号 CKref 的不同抽头 116，则可以容易地识别上升沿与反馈时钟信号 CKf 的上升沿最接近的基准时钟信号 CKref 的抽头。一种方法是，使 DLL 继续推进反馈信号 CKf 的上升沿，即通过逐步改变控制电压 Vc 的值而增大 VCDL 延迟，并监测多个 LC 信号中哪个信号切换至 ‘1’。

[0083] 另一种方法是，直接监测 ER 和 LT 信号而不是 LC 信号。ER 和 LT 信号的组合提供了 4 种可能的状态 (00、01、10 和 11)。当反馈时钟信号 CKf 的上升沿比基准时钟信号 CKref 的上升沿早到的时间大于 T2，该状态为 ‘10’ (ER 信号是 ‘1’ 且 LT 信号是 ‘0’)。当反馈时钟信号 CKf 的上升沿比基准时钟信号 CKref 的上升沿晚到的时间大于 T1，该状态为 ‘01’ (ER 信号是 ‘0’ 且 LT 信号是 ‘1’)。因此，如果反馈时钟信号 CKf 的上升沿比序号为 “N” 的 VCDL 抽头 116、116a 上的基准时钟信号 CKref 的上升沿要晚，但反馈时钟信号 CKf 的上升沿比序号为 “N+1” 的抽头 116、116a 上的基准时钟信号 CKref 的上升沿要早，那么抽头 “N” 上的状态 (ER 和 LT 信号的组合) 是 “11” 或 “01”，而抽头 “N+1” 上的状态 (ER 和 LT 信号的组合) 是 “10” 或 “11”。如果值 T1 和 T2 都被选为相邻抽头之间的时间延迟的小部分，例如，两个相邻抽头 N、N+1 之间的时间延迟的十分之一，那么 “11” 的组合不会同时出现在两个相邻的抽头上，而且容易看出的是锁定点 (状态 ‘11’) 处于这两个抽头之间。

[0084] 仅使用从每一级 118 输出的单一 LC 信号的方法比使用 ER 和 LT 信号的方法的实现更为简单。然而，使用 ER 和 LT 信号组合的方法是使 DLL 达到锁定点的更快的方法。本领域技术人员可以理解，在备选实施例中，所有三个信号 LC、LT 和 ER 的组合也可以用于对 DLL 工作点进行初始化。

[0085] 回到图 5，DLL 初始化控制 112 控制初始化过程。初始化控制 112 是状态机。该过程包括在 DLL 初始化过程期间为所施加的信号 (RST、HLD、MX) 选择正确的值。本领域技术人员可以理解，存在多种方式来实现或合成状态机。因此，对用于合成具有期望的 DLL 初始化控制 112 的功能的状态机的算法进行描述，而不描述 DLL 初始化控制自身的内部结构的示意图或元件。还要注意的，这里讨论的且在图 11 中所示的算法不是可用于实现初始化过程的唯一逻辑方案。步骤的次序可以改变，而且可以增加、去除或修改步骤。

[0086] 图 11 是示出了 DLL 初始化控制 112 中实现的 DLL 初始化过程的流程图。每当发生上电或系统复位时，开始 DLL 初始化过程。结合图 9 和图 5 来描述图 11。

[0087] 在步骤 200，参考图 5，把系统复位信号 (为了简便起见未示出) 耦合至 DLL 500 和 DLL 500 内的各个模块。把 RST 信号输入锁定检测器 111 和控制电压发生器 107。如果需要，RST 信号对锁定检测器 111 进行复位。例如，当把复位信号耦合至触发器 119a、119b 的清除输入端时，通过对图 9 所示的触发器 119a、119b 进行清除以使每一个触发器的各个 Q 输出端被设置为 ‘0’，从而对锁定检测器 111 进行复位。复位信号还把控制电压节点 Vc 放

电至边界值,即产生最小延迟的值。复位信号还断言 HLD 信号为“保持”模式,设置 MX 控制代码 152 的初始值,并设置被搜索的锁定点的期望临近度。

[0088] 在上电时,把 MX 代码值 152 设置为初始值,从而选择具有最小相移的 VCDL 抽头输出 116 或 116a(图 7 和 8 所示的最左侧的抽头输出(相位 1))。通过选择锁定检测器 111 而不是 PD 104 作为用于估计时钟信号(CKref、CKf)上升沿的临近度的设备,对期望的锁定临近度进行设置。

[0089] 在上电时来自 DLL 初始化控制 112 的 MX 输出 152 的值取决于实施方式而不同。MX 输出 152 可以是单比特或多比特。在图 5 所示的实施例中,MX 是多比特代码。稍后讨论单比特和多比特的实施方式。当把 HLD 信号设置为“保持”时,PD 104 处于“保持模式”且 PD 104 控制 UP 和 DOWN 输出信号,从而 VCDL 102 的延迟仅能够增大。当 PD 104 处于“保持模式”时,其分离的专用输出信号仍能够用于以更高的精确度测量输入时钟信号(CKref、CKf)上升沿的临近度。

[0090] 即使是最精确的相位检测器,也只能以特定的有限精确度进行边沿对齐。这个精确度尤其确定了 DLL 时钟对齐的精确度。例如,如果 PD 误差是 20ps 且 DLL 处于稳定锁定,且排除其他因素,则时钟边沿彼此处于 20ps 内。因此,当时钟边沿之间的时间差为 20ps 或更小时,PD 输出信号指示时钟边沿对齐。

[0091] DLL 的初始化以最小 VCDL 102 延迟开始,而且延迟只能增大。回到图 11,在步骤 202,时钟(CKref、CKf)前进 1 个周期,(即一个时钟周期),如之前讨论的,通过锁定检测器和 / 或 PD 104 来测量各个时钟周期中的时钟(CKref、CKf)的上升沿对齐。

[0092] 在步骤 204 处,DLL 初始化控制 112 根据从锁定检测器 111(图 5)转发来的 LOCK 代码 154(图 5),确定是否已经以期望的临近度达到锁定。锁定临近度是查找锁定点的精确度。锁定临近度存储在 DLL 初始化控制 112 中,作为寄存器中的可加载代码或熔丝编程代码。锁定临近度用于对锁定检测器 111 中的 T1 和 T2 时间间隔的值进行设置。

[0093] 如前所述,锁定检测器 111 可以确定粗略对齐或精确对齐,即上升沿的临近度(粗略或精确),临近这由 T1 和 T2 时间间隔值的设置以及是否将 ER、LT、LC 信号或其组合用于选择锁定点来确定。

[0094] 对齐精确度可以在 DLL 初始化过程中改变。在初始化过程的开始,粗略地检测对齐是足够的。典型地,正如结合图 9 和 10 所述的那样,该粗略对齐由锁定检测器 111 来执行。在 DLL 初始化过程开始时,当适合的边沿对齐令人满意时,使用具有长的 T1 和 T2 时间间隔和简单的 LOCK 代码的锁定检测器级 118。当需要更加精确的对齐时,使用更短的 T1 和 T2 时间间隔和 LOCK 代码中更加复杂的 ER、LT、LC 信号组合。备选地,PD 104 可以参与到这个级中,作为边沿对齐检测器。

[0095] 继续执行初始化过程:在每一个上升时钟沿,DLL 估计时钟信号上升沿之间的临近度(对齐)并产生校正信号。根据来自相位检测器 104 的 UP、DOWN 信号的值,对控制电压 Vc 不断进行修改。

[0096] 在若干最初的时钟周期后,LOCK 代码指示 DLL 逼近第一锁定点。在以期望的临近度达到第一锁定点后,过程继续执行到步骤 206。

[0097] 在步骤 206,通过修改 MX 代码从而通过多路复用器 113 对反馈时钟信号 CKf 进行移相,以跳过第一锁定点。通过在上电或复位后很少的时钟周期后执行移相,可以快速增大

VCDL 延迟。在这种情况下,使用具有高粒度 VCDL 抽头的精确锁定检测器,它允许对第一和第二锁定点 604、606 进行快速和精确定位,这又允许快速产生与第二锁定点相对应的 MX 代码的值。

[0098] 参考图 6B 的时序图和图 6A 的图表,在达到并跳过第一锁定点 604 后,通过修改控制电压 V_c 的值使延迟继续增大,从而 DLL 移向下一个、第二锁定点 606。该延迟最初通过移相而增大,之后通过逐步修改控制电压 V_c 而逐步增大,即在每一个时钟周期做出相对小的改变。该时序图还示出通过逐步增大 VCDL 延迟而达到第二锁定点 604。控制电压 V_c 的值的逐步改变比相移要慢,然而仍旧会增大 VCDL 延迟,并允许 DLL 达到与第二锁定点相对应的控制电压 V_c 值。

[0099] 利用被设置为与 VCDL 最小延迟相对应的值的控制电压 V_c ,可在单次相移中使 DLL 达到最终锁定点(例如第二锁定点 606)。为了执行该操作,选择抽头 116、116a,使得 DLL 跳过第一锁定点并达到与第二锁定点很近的地方,其中保持 V_c 产生最小延迟。然而,在当工作条件发生改变从而释放 V_c 后,DLL 将不会有太多空间来减小延迟,因为控制电压 V_c 离产生最小延迟的值不是太远。因此,DLL 将容易失去锁定,需要明显地减小延迟以补偿时钟信号边沿的漂移和工作条件的改变。

[0100] 由于 V_c 电压电平的修改仅需大于或至少等于 DLL 正常操作期间所需的最大 VCDL 延迟减小(图 3B 中的范围 304)加上一些安全富余,所以在最终锁定前可以实现 VCDL 延迟增大的一部分。明显地,选择更长的定时间隔将使针对最终锁定的搜索也变得更长。然而,如上所述,该定时间隔由最大 VCDL 延迟修改(减小)来确定,在时钟信号边沿漂移和工作条件改变的影响下,在正常的 DLL 操作期间的任意时刻可能需要该修改。

[0101] 在一个实施例中,锁定检测器 111 包括多个级 118,其中仅有锁定检测器 111 中使用的每一个级 118 的 LC 输出才会输出 LOCK 代码。这个实施例中的级 118 具有嵌入其各自延迟线 122a-c 中的不同的 T1 和 T2 延迟时间值。这种情况下的延迟线 122a-c 是 VCDL 102 缓冲器链 114 的一部分的拷贝。第一级 118a 在延迟线 122a-c 中具有较少数目的缓冲器,而且当基准时钟信号 CKref 和反馈时钟信号 CKf 的上升沿彼此更近时,断言下一级的 LC 信号(设置为逻辑‘1’)。

[0102] 每一个下一级 118b 相比于前一级具有更多的缓冲器。因此,每一个下一级 118b 的临近度更不精确(更粗略),而且当时钟信号(CKref、CKf)的上升沿进一步彼此远离时,断言相应 LC 输出(设置为逻辑‘1’)。从所有级输出组合的 LC,即在这个实施例中是锁定检测器 111 的 LOCK 输出代码。LOCK 代码指示需要向 VCDL 中基准时钟信号 CKref 的路径中添加多少个单一 VCDL 缓冲器,以便使时钟信号 CKref 和 CKf 的上升沿彼此接近。换句话说,LOCK 输出代码 154 标识由 MX 代码 152 应从抽头 VCDL 输出中选择的抽头,以便跳过第一锁定点 604,从而使 DLL 工作点与第二锁定点 606 接近。

[0103] 因此,在步骤 208,DLL 初始化控制 112 通过断言与第二锁定点相对应的 MX 代码值并切换至 PD 104 控制,将期望的临近度变为精确的。在初始化过程的这个阶段,使用专用的 UP 和 DOWN 信号来以高精度估计精确度时钟上升沿定位,同时 UP 和 DOWN 输出仍由 HLD 信号来保持,以便仍旧仅增大 VCDL 102 延迟。备选地,锁定检测器 111 可以保留受控,同时保持 PD 104 产生 UP 和 SOWN 输出信号以增大 VCDL 102 的延迟时间。

[0104] 在步骤 210,时钟(CKref、CKf)前进一个周期。检测基准时钟信号 CKref 和反馈

时钟信号 CKf 的下一上升沿,并测量临近度。

[0105] 在步骤 212,如果以期望的临近度达到第二锁定点,则过程继续到步骤 214。如果不是,则过程回到步骤 210 以继续监测基准时钟信号 CKref 和反馈时钟信号 CKf 之间的对齐度。

[0106] 在步骤 214,达到更加精确的最终锁定,而且通过把 HLD 信号切换至“释放”状态而释放 PD 104。

[0107] 在步骤 216,PD 104 控制控制电压 Vc,而且正常的 DLL 操作开始。

[0108] 图 12-15 中示出了 DLL 的其他实施例。

[0109] 图 12 是根据本发明原理包括 DLL 初始化的 DLL 140 的备选实施例框图。这个实施例更适于需要更快的 DLL 初始化的情况,即当达到最终 DLL 锁定点且切换至正常 DLL 操作所需的时间是关键时。

[0110] DLL 140 具有多个与结合图 5 所述的 DLL 500 的实施例共同的元件。正如在图 5 所示实施例中所讨论的,DLL 包括具有多个抽头输出 116 或 116a 的 VCDL 102、PD 104、控制电压发生器 107、多路复用器 113 以及 DLL 初始化控制 112。与图 5 所示 DLL 500 中的单一拷贝延迟 103 不同,DLL140 包括多个拷贝延迟块 103,VCDL 102 抽头输出 116 或 116a 中的每一个都对应一个拷贝延迟块 103。

[0111] 锁定检测器 111 具有图 9 所示的内部结构,具有与 VCDL 抽头输出 116 或 116a 的数目相等数目的多个级 118。把所有级 118 中的定时间隔 T1 和 T2 设置为相同的值。定时间隔 T1、T2 的值远小于两个相邻 VCDL 抽头 116 或 116a 之间的延迟时间。锁定检测器 111 输出的 LOCK 代码包括所有级 118 的 LC、LT 和 ER 输出。这允许以更高的精确度确定时钟信号上升沿的相互位置。使用两个多路复用器 113a-b:多路复用器 113a,用于从多个 VCDL 102 抽头输出中选择 DLL 输出时钟信号;以及多路复用器 113b,用于选择反馈时钟信号 CKf。两个多路复用器 113a-b 由相同的 MX 多比特代码来控制。与结合图 5 所述的实施例相比较,需要额外的硅面积开销以容纳这两个多路复用器 113a-c、多个拷贝延迟 103 以及锁定检测器 111 中的锁定检测器级 118,以提供更高的精确度和更快的锁定时间。

[0112] 来自抽头 VCDL 输出 116 或 116a 的时钟信号被转发到拷贝延迟 103 和第一多路复用器 113a。在经过拷贝延迟 103 后,时钟信号被转发到锁定检测器 111 和第二多路复用器 113b。DLL 初始化控制 112 控制结合图 11 所述的初始化过程。在上电或系统复位时,控制电压发生器 107 把控制电压 Vc 复位至边界值,从而 VCDL 102 产生最小延迟。为了简便起见,去除了图 5 的实施例中所示的复位信号。HLD 信号值的状态被设置为“保持”,以保持 PD 104 处于延迟增大模式,锁定检测器 111 被复位,而且 MX 多比特代码此时要么被设置为缺省值,要么仍处于初始随机值。

[0113] 通过允许锁定检测器 111 对控制电压 Vc 进行控制并保持 PD 104 处于“保持”模式,来设置期望的临近度。在若干时钟周期后,LOCK 代码值根据拷贝延迟 103 的输出端上多个时钟信号的上升沿相对于基准时钟信号 CKref 的上升沿的位置,以高精度稳定并指示时钟信号 CKref、CKf 之间的相位差。使用从锁定检测器 111 接收到的 LOCK 代码值,DLL 初始化控制 112 通过多路复用器 113a 为输出时钟信号 CKout 选择 VCDL 102 输出的最接近的时钟,并通过多路复用器 113b 选择拷贝延迟单元 103 输出的相应时钟信号 CKf 以反馈至 PD 104。

[0114] 为了确定 MX 代码的值,重要的是注意,由两个多路复用器 113a、113b 所选择的时钟信号涉及 VCDL 102 的相同抽头输出。因此,时钟信号由 MX 代码来选择,而几乎是在复位或上电后直接执行移相。在移相后,DLL 的工作点接近于第二锁定点 606(图 6A),而且 DLL 初始化控制 112 通过 PD 104 逐步增大控制电压 V_c ,同时 HLD 信号保持 PD 104 的输出,从而 PD 104 仅增大 VCDL 延迟。控制电压 V_c 继续增大,直到达到第二锁定点 606(图 6A)。在达到第二锁定点 606(图 6A)后,DLL 初始化控制 112 把 HLD 信号的状态改变为“释放”,而且通过把对控制电压 V_c 的控制转移给 PD 104 而把期望的锁定临近度切换至“精确”。DLL 的正常操作开始。即使在这个“加速锁定”实施例,也不能在 V_c 维持与最小延迟相对应的值时仅通过相移而使 DLL 精确地达到第二锁定点。如上所述,需要针对 VCDL 延迟变化的富余,对时钟边沿漂移和工作条件改变进行补偿。因此,移相使 DLL 与第二锁定点接近达到所述富余,并通过逐步修改 V_c 来覆盖余下的路线(所述富余)。

[0115] 图 13-15 所示的 DLL 的实施例包括先前结合图 5 所述的 DLL 的实施例或结合图 1 所述的现有技术的 DLL 中未使用的反相单元 110。

[0116] 参考图 13,在最简单的情况下,反相单元 110 包括反相器 109 和多路复用器 108。反相单元 109 的功能是把输入信号分为反相的和非反相的形式,并根据单比特 MX 信号的逻辑值而选择一个形式以提供至输出端。例如,当 MX 信号是‘0’时,输出非反相的信号,当 MX 信号是‘1’时,输出反相的信号,或者相反。因此,反相单元 110 与先前结合图 12 所示实施例而描述的多路复用器 113 类似。

[0117] 如本领域技术人员所公知的,存在多种可能的方式来实现反相单元 110 的功能。可以选择能够提供该单元功能的任意变体。本领域技术人员可以理解,反相单元 110 的内部结构可以和图 13 所示的不同。例如,反相单元 110 可以具有单端或差分输入端以及差分输出端,具有由 MX 信号所选择的相位或相反相位输出端。反相单元 110 还可以包括附加链,用于平衡该单元内两个路径——反相和非反相——的延迟。还可以理解的是,在 VCDL 102 输出提供信号和补码对 117a(图 8)或信号和补码 117 以及 117’(图 7)的情况下,反相器 109 可不必在反相单元 110 中。在这种情况下,反相单元 110 包括 2 输入和 1 输出形式的多路复用器 113。

[0118] 图 13 所示的 DLL 的实施例不会像结合图 12 所述的实施例那样快速地查找锁定点。然而,图 13 中的实施例实现起来更简单且需要更少的硅面积开销。

[0119] DLL 142 仅包括一个 VCDL 输出、一个反相单元 110 和仅具有一个级 118 的锁定检测器 111。备选地,在 VCDL 102 如图 8 所示提供输出信号及其补码 117a 的情况下,反相单元 110 可以被 2 输入和单输出的多路复用器 113 所取代。锁定检测器 111 中的定时间隔 T1 和 T2 被设置为时钟周期的小片段。锁定检测器 111 仅当基准时钟信号 CKref 和反馈时钟信号 CKf 的上升沿彼此接近时才检测锁定点。

[0120] 在复位或上电后,DLL 初始化控制 112 从初始延迟开始逐步增大延迟。在根据锁定信号的状态而检测到第一锁定点 604(图 6A)后,反相单元 110 根据 DLL 初始化控制 112 输出的 MX 信号而切换至反相的时钟,从而跳过第一锁定点。在反相单元 110 执行切换后,HLD 信号仍保持 PD 104 仅通过增大控制电压 V_c 而增大延迟。DLL 142 通过根据基准时钟信号 CKref 和反馈时钟信号 CKf 的上升沿之间的时间延迟而增大控制电压 V_c ,从而继续逐步增大延迟。在检测到第二锁定点 606(图 6A)后,通过改变 HLD 信号的状态而释放 PD 104,并

且开始正常操作。

[0121] 在图 14 和 15 所示的 DLL 的实施例中,把反相单元 110 连接在与先前讨论的实施例不同的位置。

[0122] 参考图 14,DLL 144 包括两个反相单元 110a、110b。反相单元 110a 的输入端与拷贝延迟 103 的输出端相连。反相单元 110b 的输入端与 VCDL102 的输出端相连。反相单元 110a 的输出端是 DLL 输出时钟信号 CKout。反相单元 110b 的输出端是 DLL 内部反馈时钟信号 CKf,该信号被耦合至 PD 104 的输入端和锁定检测器 111 的输入端。

[0123] 参考图 15,DLL 146 包括一个反相单元 110,该单元连接在基准时钟信号 CKref 与 VCDL 102 的输入端之间,即连接在基准时钟信号 CKref 的路径中。

[0124] DLL 144(图 14)或 DLL 146(图 15)均没有示出复位信号或 MX、RST、HLD 内部 DLL 信号。为了简便起见,把这些信号从图中去除。本领域技术人员可以理解,所有这些信号出现在系统中,并以和所有先前讨论的实施例相同的方式而施加。

[0125] 本领域技术人员可以理解,使用 DLL 初始化过程的 DLL 结构不限于所述的结构。从这里关于 DLL 初始化过程所描述的思想中,有经验的设计者可以设计其他的 DLL 结构。

[0126] 在其他实施例中,DLL 通过从延迟与控制电压关系特性曲线上与最大延迟接近的点开始搜索锁定点。在初始化期间,锁定点的搜索方向被限制为仅减小延迟,即迫使相位检测器仅产生与减小延迟相对应的 UP 和 DOWN 信号的组合,这取决于相位检测器的内部结构。在一个实施例中,起始点是与最大延迟相对应的陡峭区之下的点,以避免这个区域。由于工作点将小于初始延迟,所以可以仅使用 DLL 中的某些级,其他级在初始化期间可以被禁用。

[0127] 虽然本发明参考优选实施例而得以具体示出和描述,本领域技术人员可以理解,在不背离所附权利要求包括的本发明的范围的前提下,可以在形式和细节上做出各种改变。

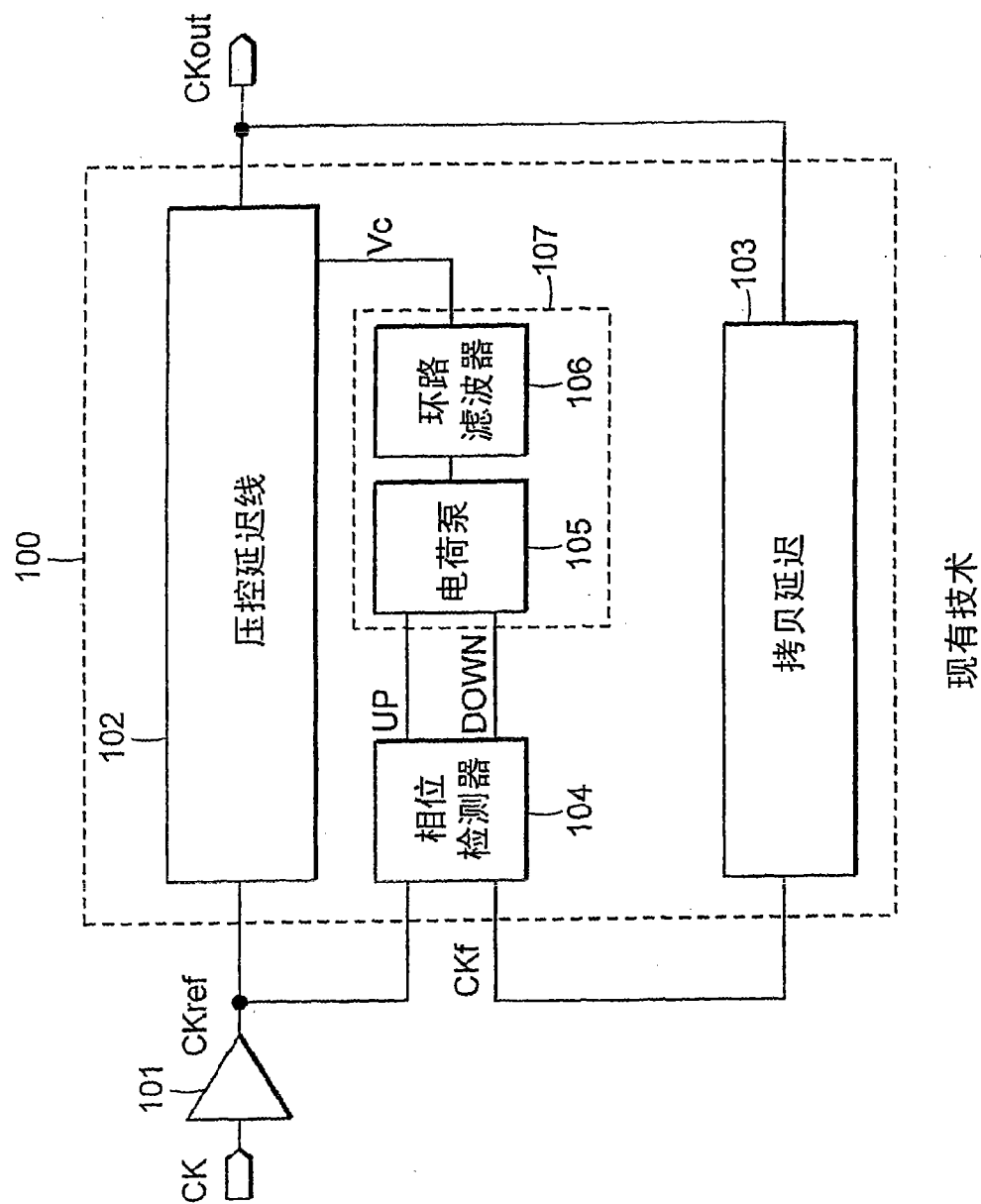


图 1

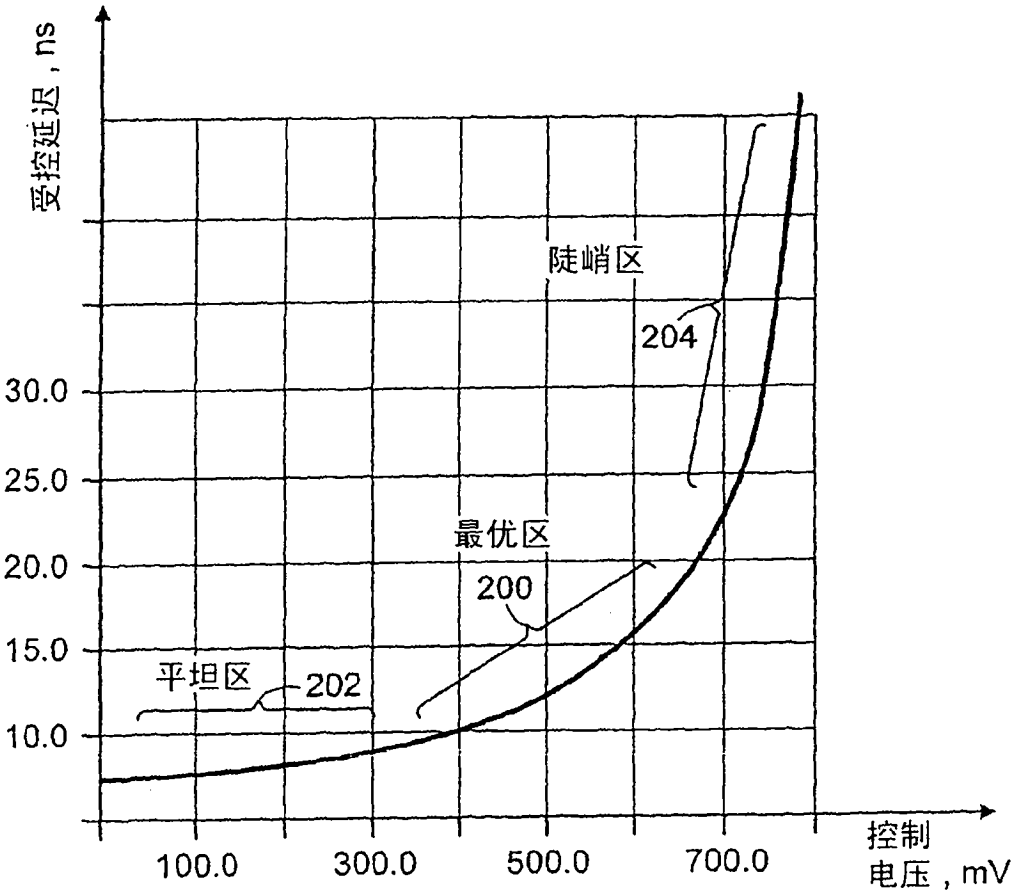


图 2

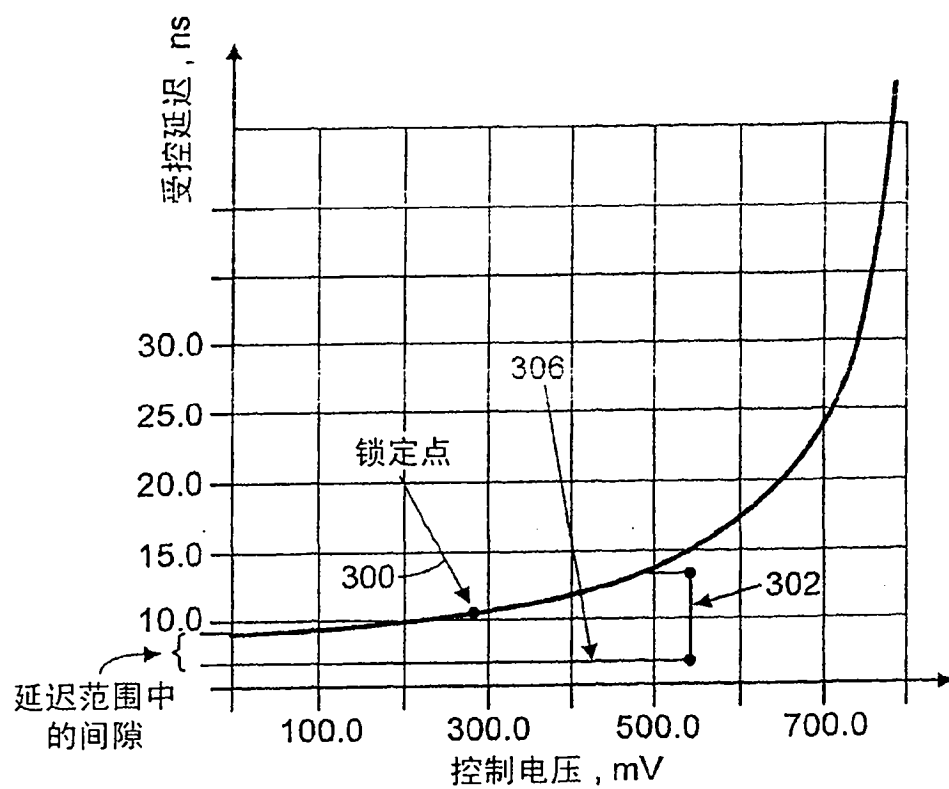


图 3A

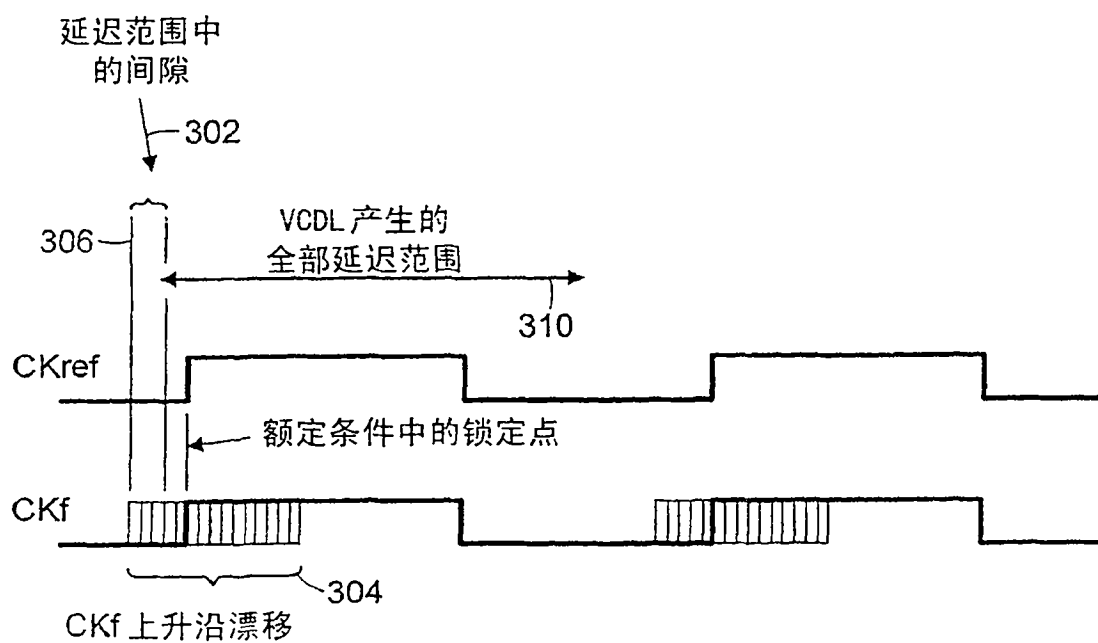


图 3B

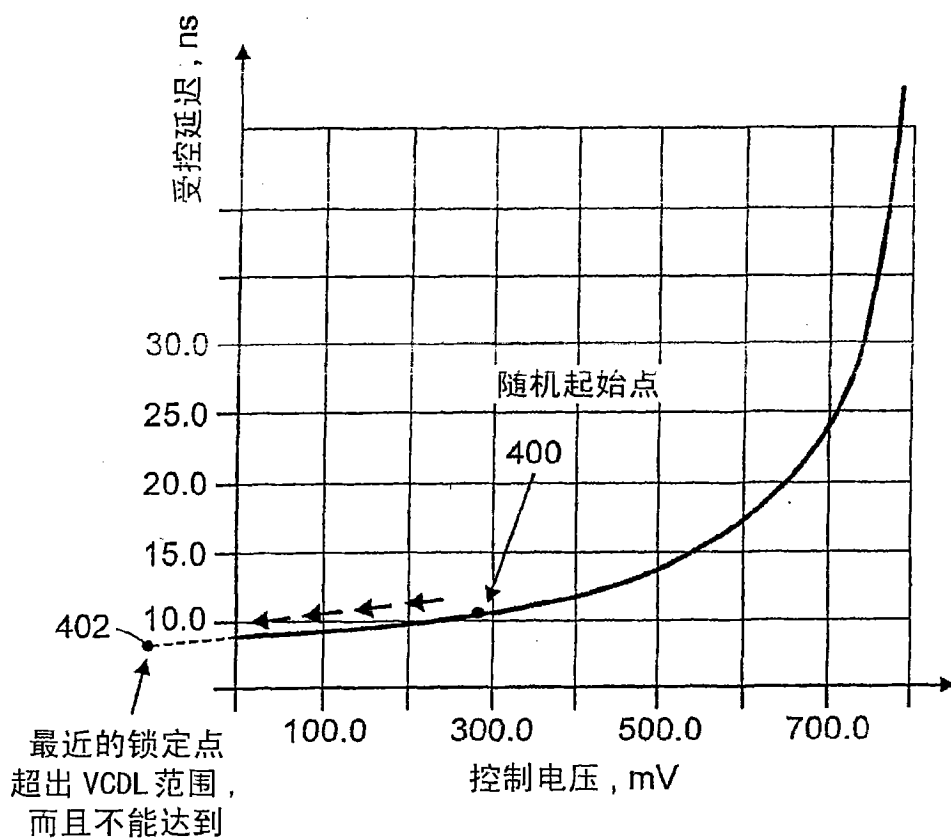


图 4A

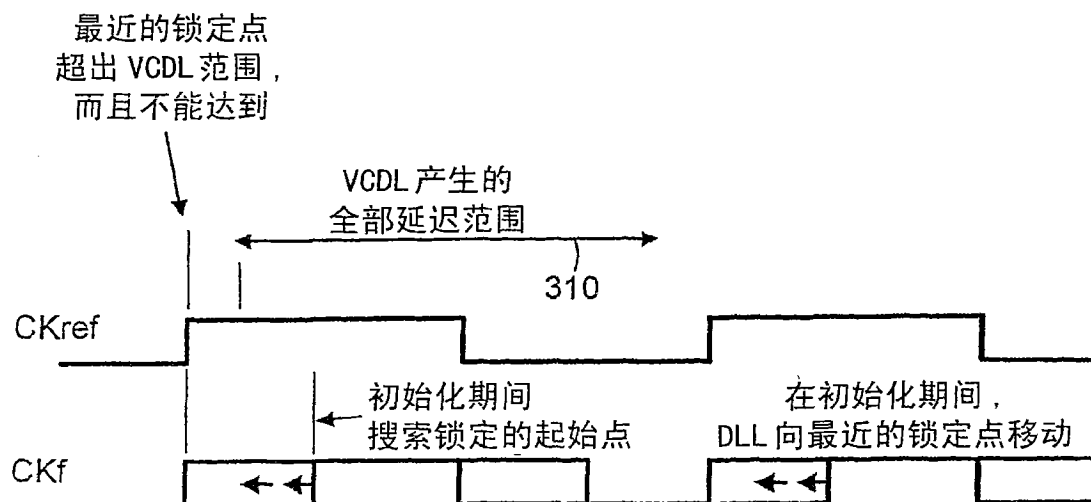


图 4B

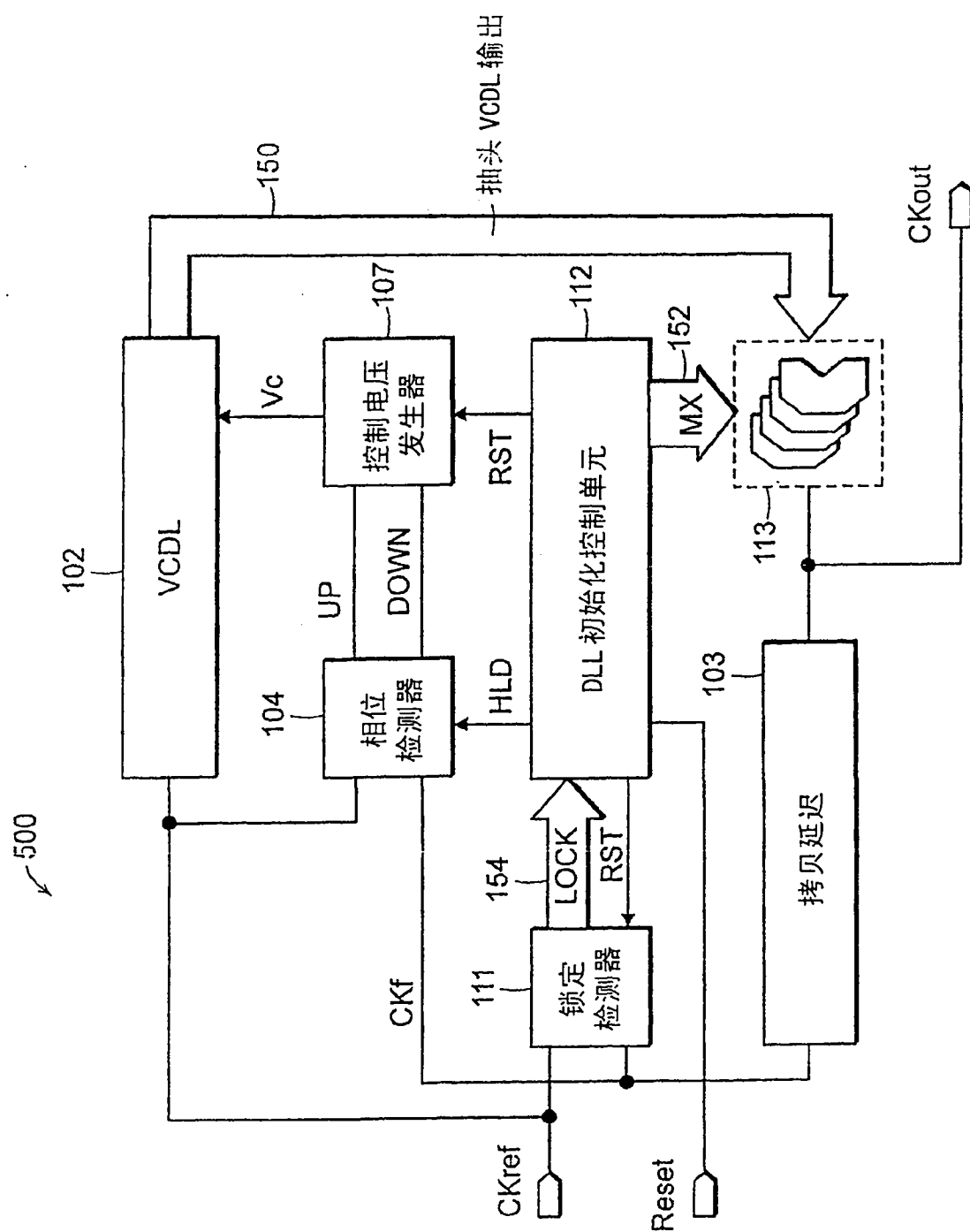


图 5

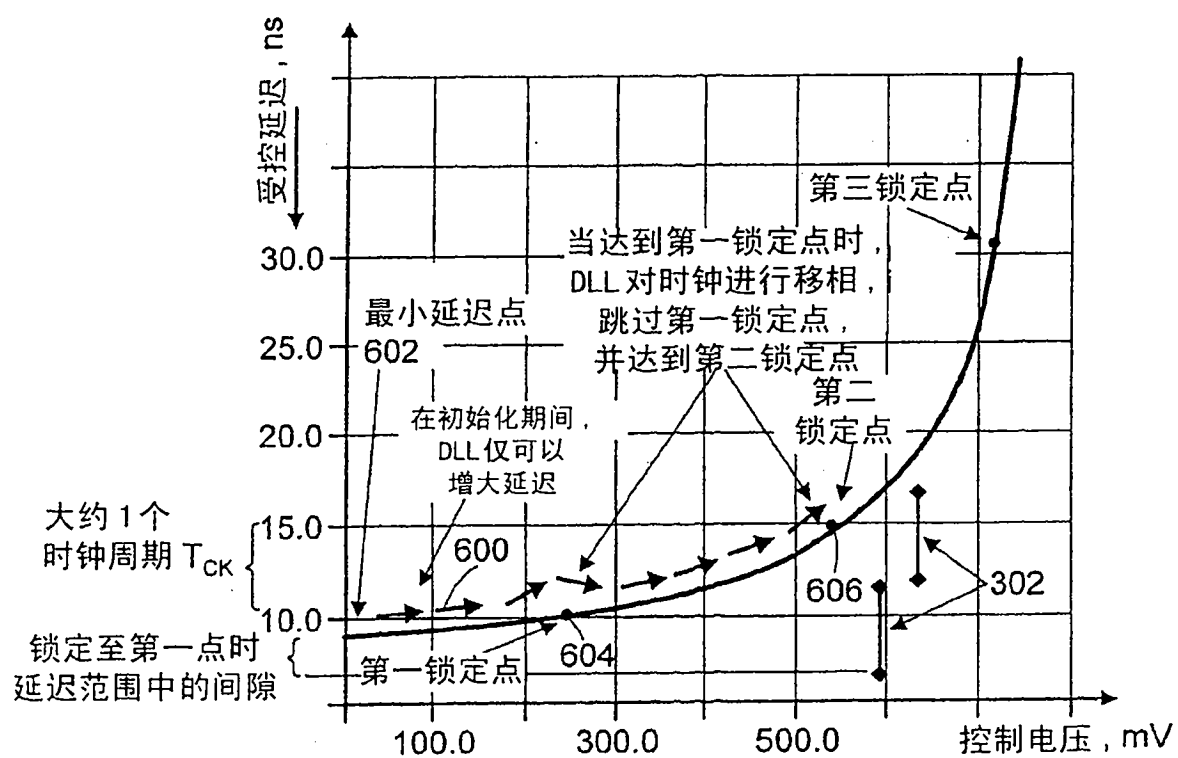


图 6A

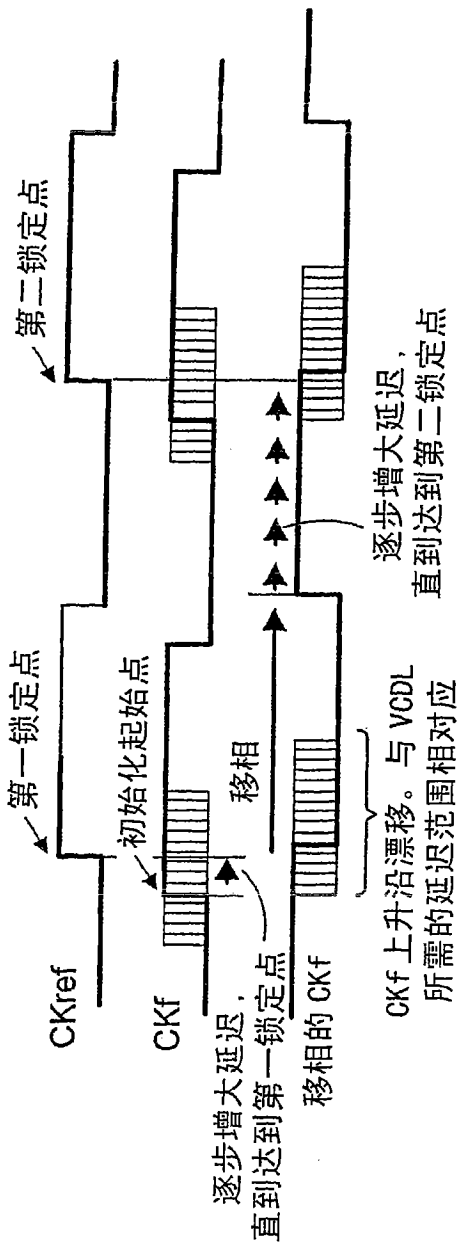


图 6B

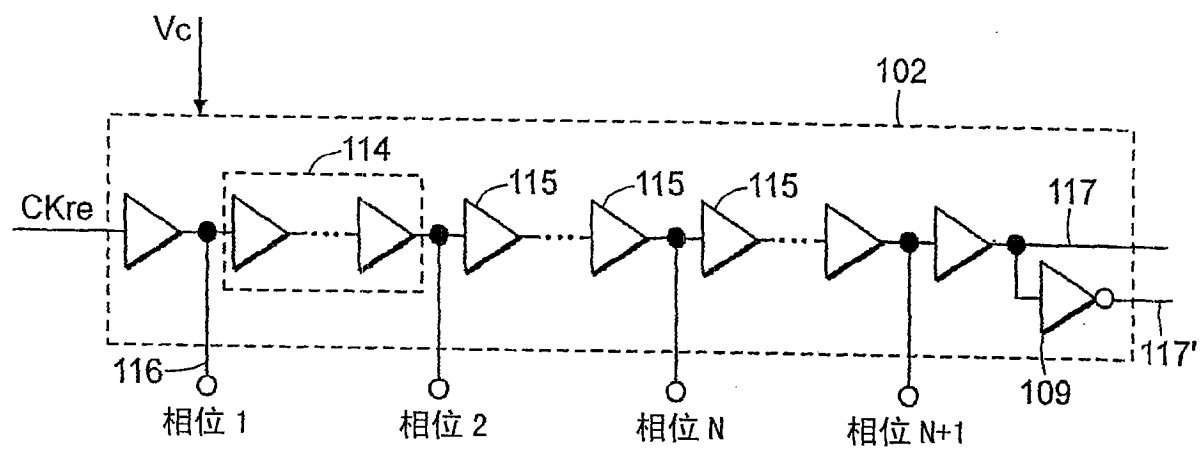


图 7

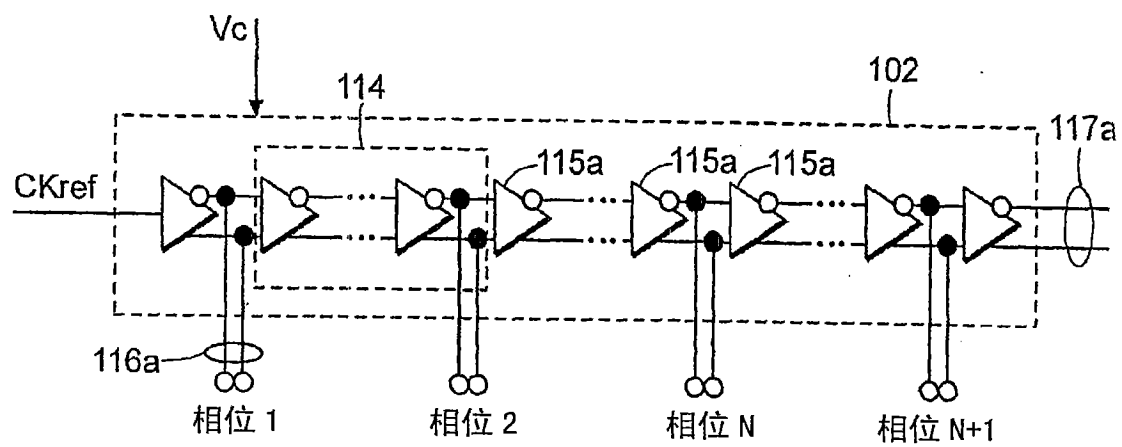


图 8

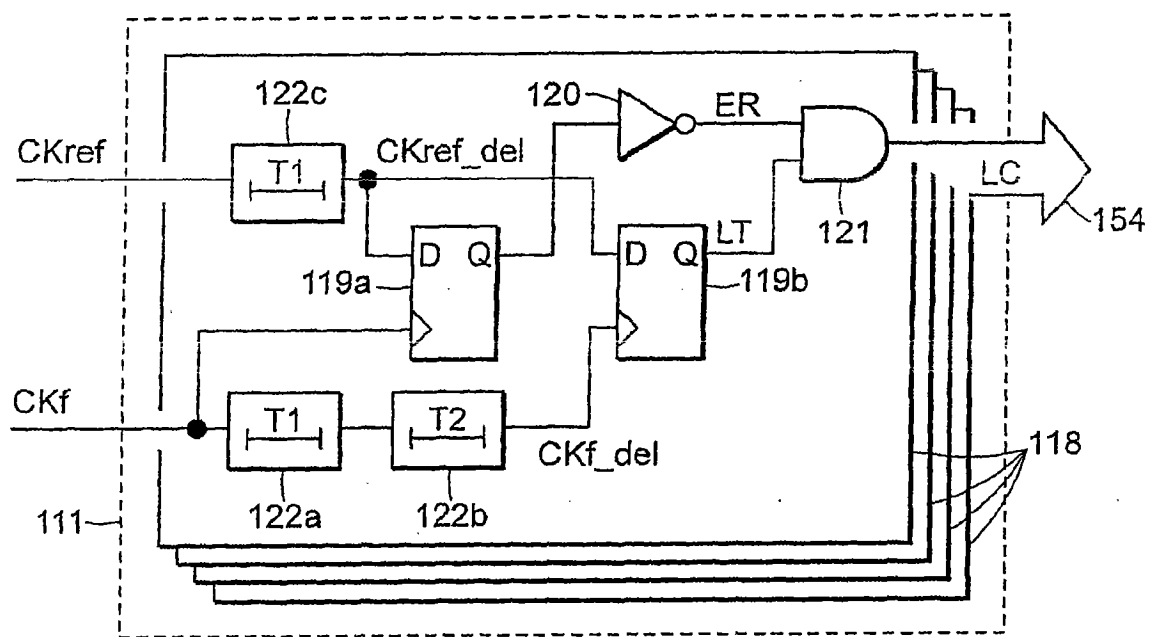


图 9

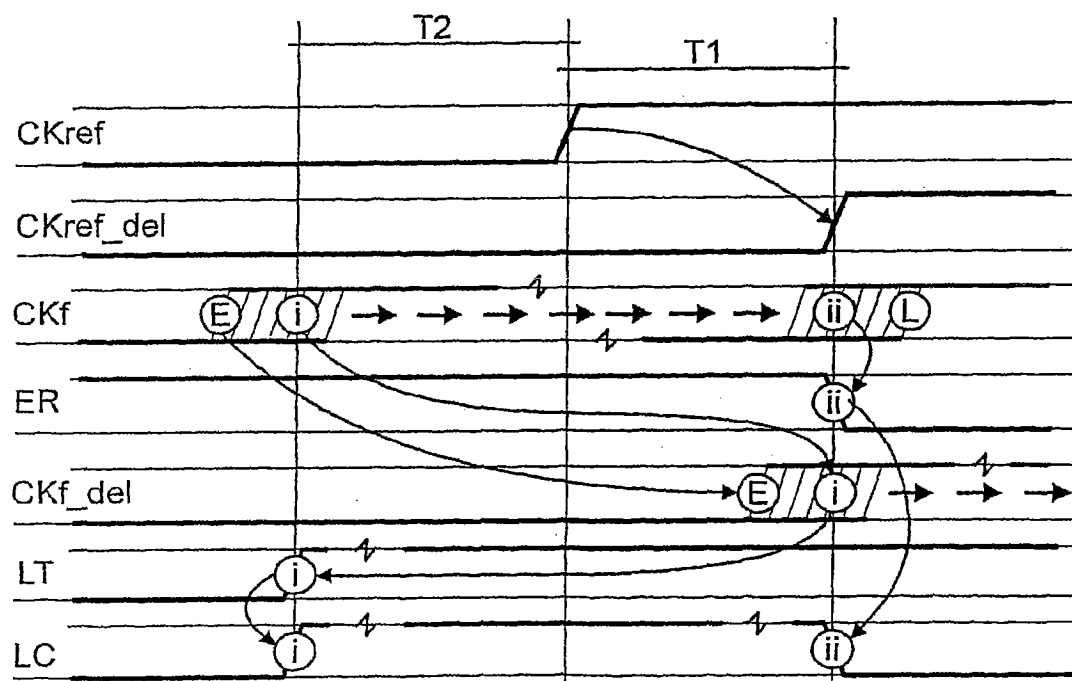


图 10

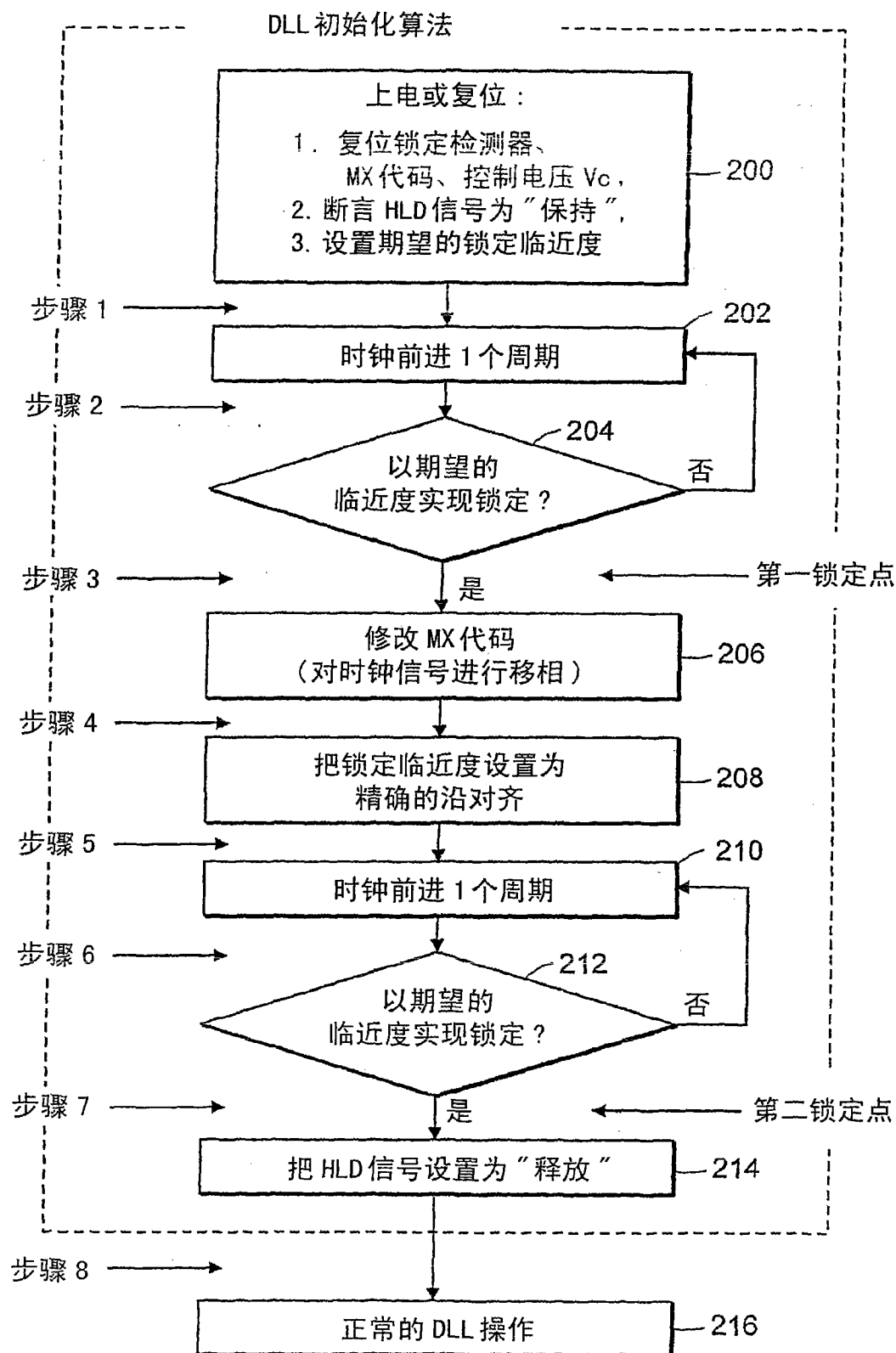


图 11

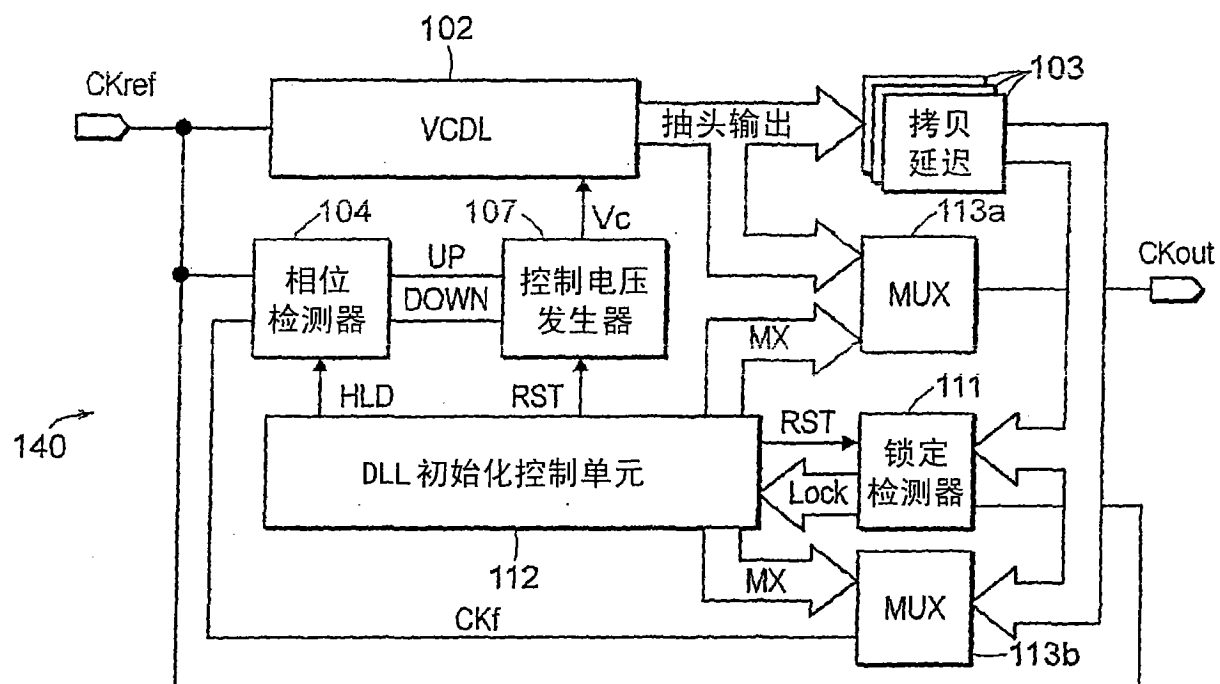


图 12

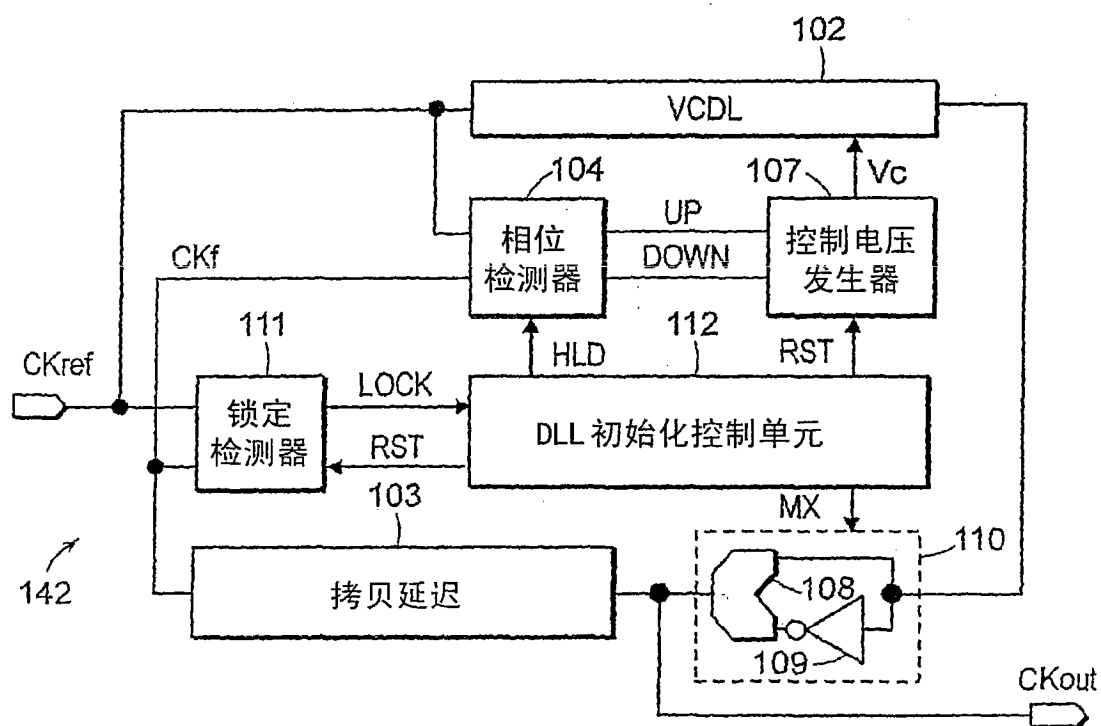


图 13

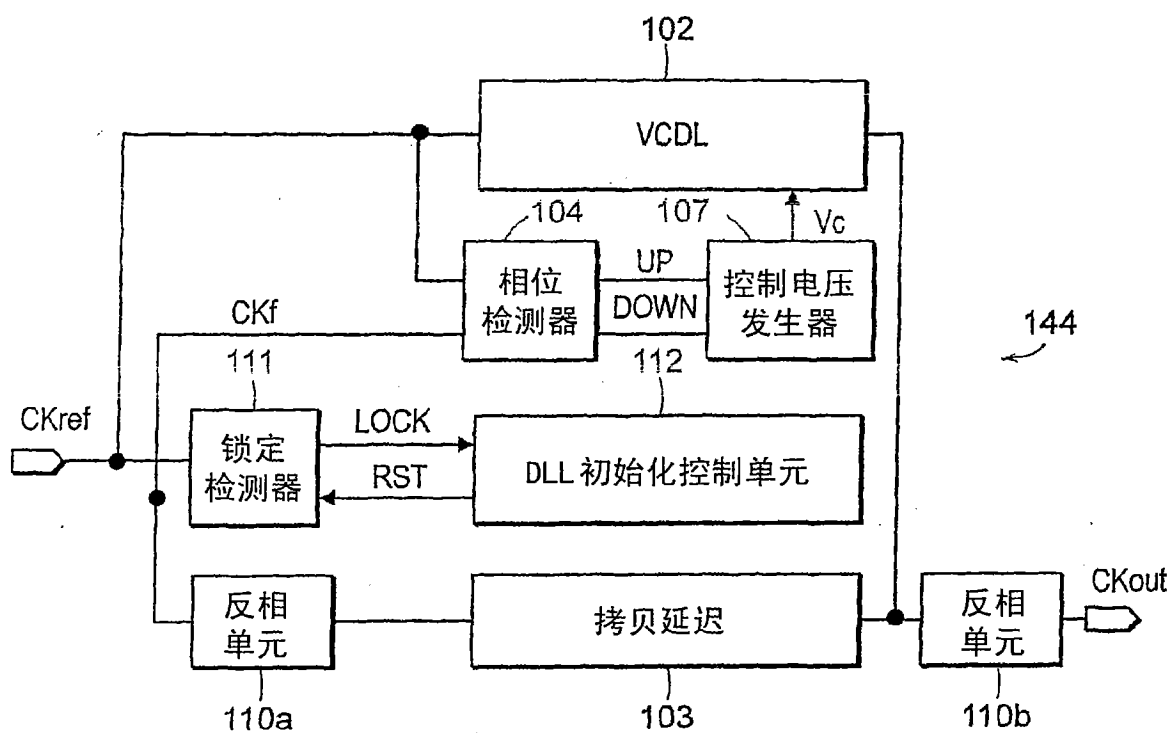


图 14

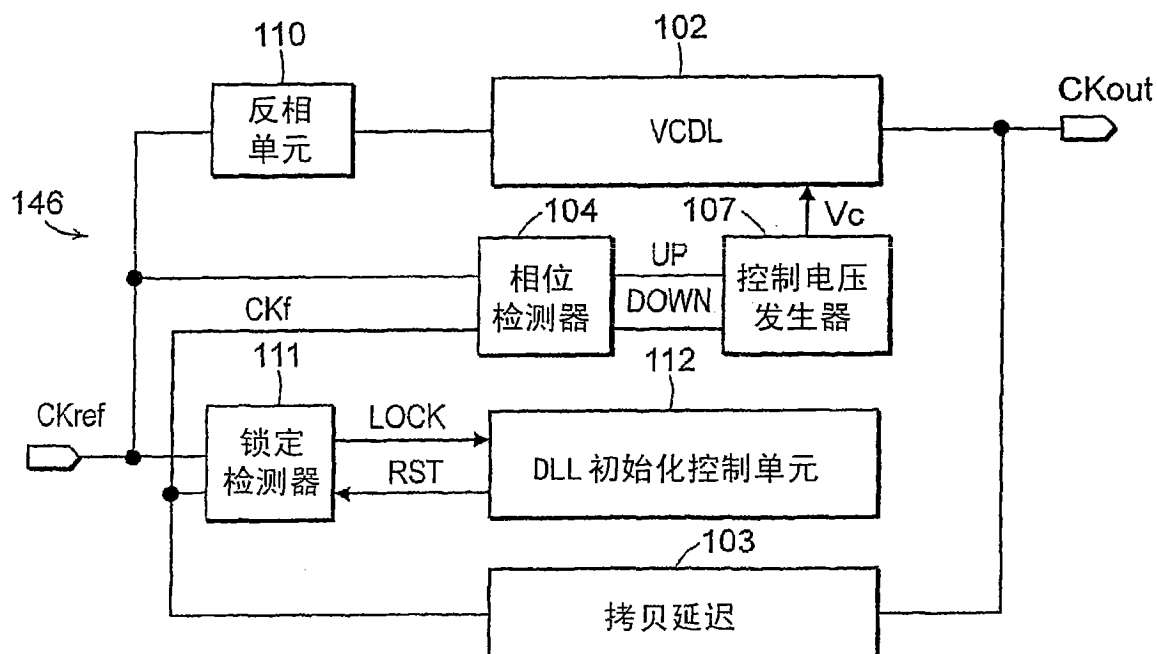


图 15