

[19] 中华人民共和国国家知识产权局

[51] Int. Cl<sup>7</sup>

G06F 13/14

G11C 11/40



# [12] 发明专利说明书

[21] ZL 专利号 96101236.6

[45] 授权公告日 2004 年 5 月 19 日

[11] 授权公告号 CN 1150462C

[22] 申请日 1996.2.9 [21] 申请号 96101236.6

[30] 优先权

[32] 1995. 2. 10 [33] US [31] 08/386,581

[71] 专利权人 汤森、汤森和克鲁

地址 美国加利福尼亚州

[72] 发明人 罗伯特·J·普罗斯汀

审查员 李婷婷

[74] 专利代理机构 中原信达知识产权代理有限责  
任公司

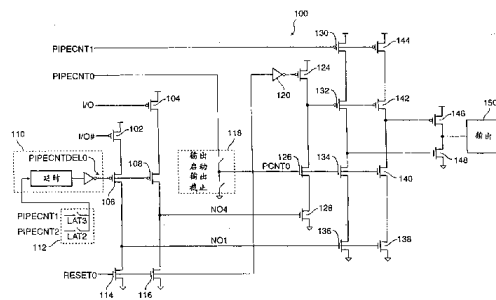
代理人 余 滕

权利要求书 15 页 说明书 48 页 附图 11 页

[54] 发明名称 用于在集成电路中流水线传送数据的方法及装置

[57] 摘要

一种在由时钟输入端所接收的第一时钟脉冲起始读周期的同步集成存储电路中使用流水线数据的方法和装置。通过存储器非同步地传播与读周期相关的数据，以产生而后输入到流水线电路的数据。该装置包括用于把在读周期所产生的数据导流到一个寄存器几个支路中的一个所择用的支路中的定时精确的导流电路，以便根据接收随后的时钟，为输出选择已存储在所择用的支路中的数据。随后的时钟脉冲是在第一时钟脉冲后出现的若干个可编程的时钟脉冲。



1. 一种具有带输出端的数据通路的集成电路,包括:  
时钟输入端,用于接收时钟信号;  
用于存储数据的具有多个分支的寄存器;  
具有连接到携带数据的数据线的输入端的导流电路,所述导流电路连接到所述寄存器,并且随后响应由所述时钟信号触发的第一信号,将所述数据导流到所述寄存器的所述多个分支中的一个已选择的分支;和  
连接到所述寄存器的分支选择电路,所述分支选择电路随后响应由所述时钟信号触发的第二信号,将存储在所述多个分支中的已选分支中的数据送到输出端,  
其中,在第一数据开始通过数据通路传播之后,但在所述第一数据到达输出端之前,第二数据开始通过数据通路传播。
2. 根据权利要求1的电路,其中所述数据为脉冲形式。
3. 根据权利要求1的电路,其中每一个所述分支进一步包括第一和第二锁存器,其中所述第一锁存器响应逻辑“1”,所述第二锁存器响应逻辑“0”。
4. 根据权利要求1的电路,进一步包括具有可寻址数据存储位置的存储器阵列,所述存储器阵列连接到所述数据线。
5. 根据权利要求4的电路,其中所述存储器阵列包括

随机存取存储器。

6. 根据权利要求4的电路,其中所述存储器阵列包括非易失性存储器。

7. 根据权利要求1的电路,进一步包括连接到所述导流电路和所述分支选择电路的计数器,所述计数器产生输出计数,所述输出计数根据所述时钟信号而变化。

8. 根据权利要求7的电路,其中:

由所述时钟信号的第一事件产生所述计数器的第一计数状态,所述第一计数状态使所述导流电路将所述第一数据导流到所述寄存器的多个分支的所述已选择分支中;

由所述时钟信号的第二事件产生所述计数器的第二计数状态,所述第二计数状态使所述导流装置将所述第二数据导流到所述寄存器的多个分支的第二已选分支中;

由所述选择电路使用所述计数器的第三计数状态来进行选择,以输出存储在所述分支中的所述第一数据;和

由所述选择电路使用所述计数器的第四计数状态来进行选择,以输出存储在所述第二已选择分支中的所述第二数据。

9. 根据权利要求8的电路,其中所述的第三计数状态出现在与所述第二计数状态相同的时钟周期上。

10. 根据权利要求8的电路,其中所述的第三计数状态出现在所述第二计数状态的若干时钟周期之后。

11. 根据权利要求7的电路,其中所述的计数器为具有时钟输入端的自启动环绕式计数器。

12. 根据权利要求 11 的电路,其中所述计数器根据加到单晶体管的栅极接线端信号沿而改变输出计数。

13. 根据权利要求 12 的电路,其中所述单晶体管的栅极接线端连接到所述时钟输入端。

14. 根据权利要求 13 的电路,其中所述单晶体管连接到所述多个节点,并且通过将所述多个节点之一拉到一逻辑电平而不进行相反拉动来改变输出计数。

15. 根据权利要求 7 的电路,其中:

由所述时钟信号的第一事件产生所述计数器第一导流计数状态,所述第一导流计数状态使所述导流电路将数据导流到所述寄存器的所述多个分支的所述已选一个分支中,和

由所述时钟信号的后来的事件产生的所述计数器的第一选择计数状态,所述第一选择计数状态使所述分支选择电路进行选择,以输出存储在多个分支中的一个已选分支中的所述数据,所述后来的事件在所述时钟信号的所述第一事件后出现若干时钟周期数。

16. 根据权利要求 15 的电路,其中所述时钟周期数为 1。

17. 根据权利要求 15 的电路,其中所述时钟周期数为 2。

18. 根据权利要求 15 的电路,其中所述时钟周期数可由程序设定。

19. 根据权利要求 15 的电路,其中所述时钟周期数可由程序设定为 1 或 2。

20. 根据权利要求 15 的电路,其中所述时钟周期数可由程序设定为 0,1 或 2。

21. 根据权利要求 15 的电路,其中所述时钟周期数可大于 1。

22. 根据权利要求 15 的电路,其中所述时钟周期数可由程序设定为大于 1。

23. 根据权利要求 15 的电路,其中所述导流电路包括对应于所述寄存器的所述多个分支的多个逻辑电路,其中所述多个逻辑电路的每一个包括:

第一逻辑子电路,具有从所述数据线接收数据的第一输入端和从所述计数器导流计数状态的第二输入端,所述第一逻辑子电路对所述数据和所述导流计数状态执行逻辑 AND 或 NAND 操作;和

第二逻辑子电路,具有接收数据条的第一输入端,所述数据条为所述数据的补充,和具有从所述计数器接收所述导流计数状态的第二输入端,所述第二逻辑子电路对所述数据条和所述导流计数状态执行逻辑 AND 或 NAND 操作。

24. 根据权利要求 23 的电路,其中所述寄存器的每一个分支包括:

第一锁存器,连接到所述导流电路的相应的逻辑电路的第一逻辑子电路的输出端;和

第二锁存器,连接到所述导流电路的相应的逻辑电路的第二逻辑子电路的输出端。

25. 根据权利要求 24 的电路,其中在所述寄存器的每

一个分支中的所述第一锁存器和所述第二锁存器中的每一个进一步包括具有连接到复位信号的输入端的复位晶体管。

26. 根据权利要求 25 的电路,其中所述寄存器包括第一和第二分支,且所述导流电路包括分别连接到所述第一和第二分支的第一和第二逻辑电路,和

其中,所述第一逻辑电路接收作为输入的所述第一导流计数状态,且所述第二逻辑电路接收作为输入的第二导流计数状态,所述第二导流计数状态由所述时钟信号的第二事件在所述第一事件后产生。

27. 根据权利要求 26 的电路,所述寄存器进一步包括第三分支,且所述导流电路进一步包括分别连接到所述第三分支的第三逻辑电路,和

其中,所述第三逻辑电路从所述计数器接收作为输入的第三导流计数状态,所述第三计数状态由所述时钟信号的第三事件在所述第二事件后产生。

28. 根据权利要求 26 的电路,其中所述分支选择电路包括分别连接到所述寄存器的所述第一和第二分支的第一和第二逻辑电路,其中,每一个所述第一和第二逻辑电路包括:

第一逻辑子电路,具有接收在相应的分支中的第一锁存器的输出的第一输入端和从所述计数器接收选择计数状态的第二输入端,所述第一逻辑子电路对所述第一锁存器的所述输出和所述选择计数状态执行逻辑 AND 或 NAND 操作;  
和

第二逻辑子电路,具有接收在相应的分支中的第二锁存器的输出的第一输入端和从所述计数器接收选择计数状态的第二输入端,所述第二逻辑子电路对所述第二锁存器的所述输出和所述选择计数状态执行逻辑 AND 或 NAND 操作。

29. 根据权利要求 28 的电路,其中:

在所述选择电路中的所述第一逻辑电路,接收在所述时钟信号的所述第一事件后的出现一个时钟周期的所述第一选择计数状态,和

在所述选择电路中的所述第二逻辑电路,接收在所述时钟信号的所述第一事件后的出现二个时钟周期的所述第二选择计数状态。

30. 根据权利要求 27 的电路,其中所述分支选择电路包括分别连接到所述寄存器的所述第一、第二和第三分支的第一、第二和第三逻辑电路,其中第一、第二和第三逻辑电路每一个包括:

第一逻辑子电路,具有接收在相应的分支中的第一锁存器的输出的第一输入端和从所述计数器接收选择计数状态的第二输入端,所述第一逻辑子电路对所述第一锁存器的所述输出和所述选择计数状态执行逻辑 AND 或 NAND 操作;  
和

第二逻辑子电路,具有接收在相应的分支中的第二锁存器的输出的第一输入端和从所述计数器接收选择计数状态的第二输入端,所述第二逻辑子电路对所述第二锁存器的所述输出和所述选择计数状态执行逻辑 AND 或 NAND 操作。

31. 根据权利要求 30 的电路,其中:

在所述选择电路中的所述第一逻辑电路,接收在所述时钟信号的所述第一事件后出现的二个时钟周期的所述第一选择计数状态,

在所述选择电路中的所述第二逻辑电路,接收在所述时钟信号的所述第一事件后的出现三个时钟周期的所述第二选择计数状态,和

在所述选择电路中的所述第三逻辑电路,接收在所述时钟信号的所述第一事件后的出现四个时钟周期的所述第三选择计数状态。

32. 一种存储电路,包括:

用于存储数据的可寻址存储单元的阵列;

连接到存储单元的所述阵列的数据线;

时钟输入端,用于接收时钟信号,所述时钟信号的第一沿启动第一读周期以将输出数据提供到所述数据线上;

用于存储数据的具有多个分支的寄存器;

产生多个导流计数状态和选择计数状态的计数器,所述多个导流计数状态和选择计数状态随着所述时钟信号的周期而变化;

连接到所述寄存器和所述数据线的导流电路,所述导流电路进一步接收所述多个导流计数状态,所述导流电路将所述输出数据导流到所述寄存器的所述多个分支中的一个已选择的分支;和

由所述多个选择计数状态驱动的选择电路,用于进行选



择以输出存储在所述已选择分支中的所述输出数据，

其中，所述导流电路响应由所述时钟信号的所述第一沿触发的第一导流计数状态，将第一输出数据导流到所述寄存器的第一分支中，并且所述选择电路响应由在所述时钟信号的所述第一沿后的多个周期出现的所述时钟信号触发的第一选择计数状态，选择所述第一输出数据以进行输出。

33. 根据权利要求 32 的电路，其中每一个所述分支进一步包括第一和第二锁存器，其中所述第一锁存器响应逻辑“1”，所述第二锁存器响应逻辑“0”。

34. 根据权利要求 33 的电路，其中所述第一和第二锁存器在读周期开始处被复位。

35. 根据权利要求 32 的电路，其进一步包括禁止所述电路输出的禁止电路。

36. 根据权利要求 35 的电路，其中，当禁止所述电路输出时，防止选择每一个所述分支来进行输出。

37. 根据权利要求 32 的电路，进一步包括定时电路，该定时电路控制与在所述数据线上的输出数据脉冲的定时相关的所述多个导流计数状态的定时。

38. 根据权利要求 37 的电路，其中所述第一导流计数状态为脉冲，其持续时间比所述数据线上的所述数据脉冲的持续时间长。

39. 根据权利要求 37 的电路，进一步包括具有选通输入端和连接到所述数据线的输出端的检测放大器，其中所述检测放大器通过根据延迟信号选通来产生在所述数据线上

的所述数据脉冲。

40. 根据权利要求 37 的电路,其中所述数据脉冲为相对所述存储电路的最小周期时间的窄脉冲。

41. 根据权利要求 32 的电路,其中所述导流电路进一步包括:

等待间隔选择输入端,用以在两个等待间隔和三个等待间隔中进行选择;和

通路选择装置,用于响应所述等待间隔选择来选择用于所述导流信号的通路。

42. 根据权利要求 41 的电路,其中所述通路选择装置包括多路转接器。

43. 根据权利要求 41 的电路,其中,对于两个等待间隔选择,所述多个周期等于 1,且对于三个等待间隔选择,所述多个周期等于 2。

44. 根据权利要求 41 的电路,其中计数器周期位于三个计数状态之间,每个计数状态产生所述多个导流计数状态之一和多个选择计数状态之一。

45. 根据权利要求 41 的电路,进一步包括用于输出具有一个等待间隔的数据的附加电路。

46. 根据权利要求 45 的电路,其中当选择一个等待间隔时,禁止导流电路。

47. 根据权利要求 45 的电路,其中当选择两个或两个以上的等待间隔时,禁止导流电路。

48. 一种集成电路,包括:

适于接收时钟信号的时钟输入端；

响应所述时钟信号来产生第一、第二和第三导流计数状态和第一、第二和第三选择计数状态的计数器；

用于存储数据的具有第一、第二和第三分支的寄存器；

具有从数据线接收数据的第一输入端和分别从所述计数器接收导流计数状态的第一、第二和第三导流控制输入端的导流电路，所述导流电路连接到所述寄存器，所述导流电路响应来自所述计数器的所述导流计数状态来将数据导流到所述寄存器的所述第一、第二和第三分支中的一个已选分支中；和

具有分别连接到所述寄存器的所述第一、第二和第三分支的输出端的第一、第二和第三数据输入端，和分别接收来自所述计数器的选择计数状态的第一、第二和第三选择控制输入端的分支选择电路，所述分支选择电路响应所述时钟信号来将存储在所述第一、第二和第三分支中的已选分支的数据送到输出节点，其中所述选择的分支是响应于所述选择控制信号来选择的。

49. 根据权利要求 48 的电路，其中：

由所述时钟信号的第一事件产生第一数据，由所述导流电路响应第一导流计数状态将所述第一数据导流到所述寄存器的所述第一分支中，所述第一导流计数状态由所述时钟信号的所述第一事件触发，和

由所述分支选择电路响应第一选择计数状态来选择所述第一数据以进行输出，所述第一选择计数状态由所述时钟

信号的第二事件产生,所述时钟信号的所述第二事件出现在所述时钟信号的所述第一事件后的多个周期。

50. 根据权利要求49的电路,其中所述计数器进一步包括在输入端接收程序信息的可编程电路,且其中所述多个周期可由所述可编程电路响应所述程序信息编程设定为0、1和2中的一个。

51. 一种对出自同步存储系统的数据进行流水线传送的方法,所述同步存储系统具有带有输出端的数据通路,用以存储数据并带有多个分支的寄存器,和接收时钟信号的时钟输入端,该方法包括步骤:

在所述时钟输入端接收第一时钟沿,所述第一时钟沿启动第一数据的第一读周期;

通过所述数据通路传播所述第一数据;

将由所述第一读周期产生的所述第一数据导流到所述寄存器的所述多个分支的第一个分支中;

在所述时钟输入端接收第二时钟沿,所述第二时钟沿启动产生第二数据的第二读周期;

在所述第一数据到达所述输出端前通过所述数据通路传播所述第二数据;和

将由所述第二读周期产生的所述第二数据导流到所述寄存器的所述多个分支的第二个分支中。

52. 根据权利要求51的方法,其中所述系统进一步包括:

具有至少三个输出的环绕式计数器,在给定周期期间择

用所述三个计数器之一;和

接收禁止信号的禁止电路,所述禁止电路具有至少三个输出;

其中,除非是在择用了所述禁止信号的情况下,所述禁止电路的所述至少三个输出跟随在所述计数器的所述至少三个输出之后。

53. 根据权利要求51的方法,其中每一个所述分支进一步包括第一和第二锁存器,其中所述第一锁存器响应逻辑“1”,所述第二锁存器响应逻辑“0”。

54. 根据权利要求53的方法,其中,在将数据导流到特定分支中之前,复位所述特定分支的所述第一和第二锁存器。

55. 根据权利要求51的方法,进一步包括步骤:

在接收到所述第二时钟沿时,选择所述第一数据以进行输出。

56. 根据权利要求55的方法,进一步包括步骤:

在接收到第三时钟沿时,禁止所述第一数据输出。

57. 根据权利要求51的方法,进一步包括步骤:

接收在所述时钟输入上的第三时钟沿,所述第三时钟沿启动产生第三数据的第三读周期;

在所述第一数据到达所述输出端前传播在所述数据通路内的所述第三数据;

将由所述第三读周期产生的所述第三数据导流到所述寄存器的所述多个分支的第三个分支中;和

通过选择所述第三时钟信号来选择所述第一数据以进行输出。

58. 根据权利要求 57 的方法,进一步包括步骤:

在接收到第四时钟沿时,禁止所述第一数据输出。

59. 一种对在具有带输出端的数据通路的电路中的数据进行流水线传送的方法,该方法包括步骤:

沿所述数据通路开始传播第一数据;

在开始传播所述第一数据且在所述第一数据到达输出端前,沿所述数据通路开始传播第二数据;

将所述第一数据导流到第一存储元件;

在导流所述第一数据步骤之后将所述第二数据导流到第二存储元件;

选择在所述第一存储元件中的所述第一数据以进行输出;和

在选择所述第一数据的步骤之后选择在所述第二存储元件中的所述第二数据。

60. 根据权利要求 59 的方法,其中,将所述第一存储元件和所述第二存储元件放在距所述数据通路的输出端的基本上相同的位置。

61. 根据权利要求 59 的方法,其中,开始传播第一数据和导流所述第一数据的步骤由时钟信号的第一沿触发,并且开始传播第二数据和导流所述第二数据的步骤由所述时钟信号的在所述第一沿后的第二沿触发。

62. 根据权利要求 61 的方法,其中,所述选择所述第一

数据的步骤由在所述时钟信号的所述第一沿之后出现的多个周期的所述时钟信号沿触发,其中所述多个为0,1,2或2个以上中之一。

63. 根据权利要求61的方法,进一步包括步骤:

在开始传播所述第二数据且在所述第二数据到达输出端前,沿所述数据通路开始传播第三数据;

在导流所述第二数据的步骤后,将所述第三数据导流到第三存储元件;和

在选择所述第二数据的步骤之后选择在所述第三存储元件的所述第三数据。

64. 根据权利要求63的方法,其中,传播第三数据和导流所述第三数据的所述步骤由在所述第二沿后的所述时钟信号的第三沿触发。

65. 根据权利要求64的方法,其中,选择所述第一数据的步骤由在所述时钟信号的所述第一沿之后出现的多个周期的所述时钟信号沿触发,其中所述多个为0,1,2或2个以上中之一。

66. 根据权利要求65的方法,其中,选择所述第二数据的步骤由在所述时钟信号的所述第一沿之后出现的多个周期的所述时钟信号沿触发,其中所述多个为0,1,2或2个以上中之一,但不是选择用于所述第一数据的数目。

67. 在具有数据通路和输出端的存储电路中,一种对数据进行流水线传送的方法,包括步骤:

开始由时钟信号的第一沿触发的第一读周期,所述第一

读周期产生第一数据；

通过所述数据通路传播所述第一数据；

开始由在所述第一沿后的时钟信号的第二沿触发的第二读周期,所述第二读周期在所述第一数据到达所述输出端之前产生第二数据；

通过所述数据通路传播所述第二数据；

将所述第一数据导流到第一存储元件中；

在导流所述第一数据的步骤后,将所述第二数据导流到第二存储元件中；

选择在所述第一存储元件中的所述第一数据以进行输出,选择所述第一数据的步骤由所述时钟信号的所述第二沿触发；和

选择在所述第二存储元件中的所述第二数据以进行输出,选择所述第二数据的步骤由在所述第二沿后的所述时钟信号的所述第三沿触发。

68. 根据权利要求67的方法,其中,象所述第一存储元件一样,将所述第二存储元件放置在靠近所述数据通路的距输出端的基本上相同的位置。



## 用于在集成电路中 流水线传送数据的方法及装置

### 技术领域

本发明涉及数据处理系统,尤其涉及在一电路中的读周期被启动之后的一个选定时钟数目内用以从该电路中产生输出数据的方法及装置。

### 背景技术

在过去几年中,半导体存储器系统有了快速发展,存储器系统容量大约每四年就翻倍,每个芯片的比特数以四为倍数增加。因此,随着每次新一代的 DRAM 的出现,存储系统中的所需单个芯片数会减半。随着系统中存储器芯片数的减少,会引起数据输出引脚数目总数的下降。由于较少的输出引脚,存储器系统的带宽会下降。然而,随着微处理器和多处理器系统的进展,对于存储器系统的要求仍在提高。最要紧的问题是,计算机系统要求有越来越大的数据带宽。即,在任何给定的时间间隔上,系统要求在存储器输出缓冲单元上有更多的信息出现。所以需要增加每一个存储器芯片的带宽。由于需要保持在比特密度以及基片空间的精确性,使得实现这种更大的带宽被复杂化。

这种对于增加带宽的需求已经导致了新型存储器系统的发展。一种有前途的存储器芯片结构是“同步动态或静态随机存储器”(SDRAM 或 SSRAM)。这些芯片使用一个时钟来控制数据流并较之以原先的存储器芯片提供在输出数据带宽方面的显著增加。在这些同步设计中,流水线传送被用来增加数据输出的带宽。在本讨论中,假设通过一个示例性 SDRAM(从列地址到输出)的地址存取时间是 15ns。如果不采用流水线传送,读周期就将每 15ns 出现一次。在同步的 DRAM 中,为 3 的等待间隔(或流水线深度)可使总的速率增加两倍。就是说,对于 15ns 地址存取时间的 SDRAM,读请求和数据输出可以是每 5ns 产生一次。第一个请求的作出是在  $T_0$ 。根据该请求的数据对于 15ns(3 个时钟)之后的 DRAM 的输出是有效的。第二个读请求是在  $T+5\text{ns}$  时作出,而第三个读请求是在  $T+10\text{ns}$  时出现。出现在  $T+10\text{ns}$  处的时钟也命令从第一读周期产生的数据出现在输出端上。与在  $T+15\text{ns}$  处的第四读出请求相吻合,根据第一请求的数据可在输出端获得。该数据之后每 5ns 都跟随有出自随后读周期的新数据。结果是使系统具有比地址存取时间小得多的周期时间,显著地增加了带宽。

如果以 7.5ns 周期时间操作,这同一个 15ns 地址存取时间部分只能以一个时钟延时输出数据。即第一个读周期在  $T_0$  处开始。第二个读周期在  $T+7.5\text{ns}$  处开始,此时第一读周期的数据被命令输出。当第一数据在输出端上有效且被读出时,第三个周期在  $T+15\text{ns}$  处开始。这种操作称作为等待

间隔为2的等待间隔。此范例性的15ns地址存取时间是太慢而不能利用以5ns为周期时间正确地操作。以更高频率操作的DRAM必须使用更大的等待间隔。因而希望提供一种能力来编程特定存储器部件的等待间隔,使之以不同的操作频率数目优化其使用。正是为此考虑到流水线传送,例如在完成第一周期的存取之前即开始第二和第三周期,使得这些同步存储器较之以原先的存储器设计能够提供大大增加的带宽。

然而,遗憾的是该增加的带宽都必须付以代价才能实现。现行的流水线方案均需采用大量的传送门或锁存器以循环或“步进”数据经过该管路。通常的流水线电路是以NMOS和PMOS晶体管构成的多个传送门形成。通过使数据顺序地通过由这些传送门所定义的若干级,使这些数据钟控穿过流水线电路。因此,当等待间隔为2时,可采用两套传送门以步进这些数据经过该管路。第一套门被启动以推动在输入缓冲器处于所接收的数据经过存储器。当在输入有新数据开始时,在数据路径中定位靠后的第二套门被启动,将数据推进到输出。与之相似,等待间隔为3时要求有三套传送门,放置在经过每一并行数据路径的几个位置上。

虽然此方案达到了从同步电路流水线传送输出数据的一般目标,但仍有几个问题不令人满意。此种方案所需的大量的传送门对于几种RAM设计的特征有不利影响。例如,在SDRAM中的中间的传送门可优化定位在存储器每一列的解码器的输出。这些存储器有大量的列解码器输出。所以

需要大量的传送门以产生例如3次等待间隔。这些传送门的每一个都要耗能、占据可贵的基片面积并附加电阻和电容于电路中,从而增加了对地址存取时间的延时。结果是,利用共同流水线传送技术的同步 DRAM 电路占据了比小容量非同步电路要实际上更大的基片空间。而且,采用这种共用技术,虽然是实现了降低循环时间的目标,但其产生了存储器装置对能源要求和地址存取时间上的增加。

当采用可编程等待间隔时,本已属大量的传送门会进一步增加。例如,如果是利用传送门来实现2或3次的可选性等待间隔,对于不同的等待间隔,在整个数据路径中的门的最佳放置是不同的。以此方案会使基片的空间急剧消耗。

因此,需要能够减低或消除在数据路径中对于众多的传送门的需求的离速流水线传送技术。还希望有支持一种任何期望值的可编程等待间隔的技术。这些需要应该在有比较、密度、基片面积及能耗效益方面的严重损失条件下得以实现。

### 发明内容

本发明提供了一种流水线电路,实现了一种可编程等待间隔,它可以用很高的频率流水线传送数据而不占据过量的基片面积。

具体而言,本发明提供一种用于同步系统中流水线传送数据的方法和装置。在一个具体的实施例中,该装置设计来与一个存储器系统结合使用,其中的读周期由在时钟输入端

所接收的第一个时钟启动。该读出周期非同步地产生随后将被输入到流水线电路中的数据。该电路包括导流电路,用于将读周期内产生的数据导流进入寄存器若干支路所择用之一。选择电路被用来在一旦出现随后时钟的接收时从所择用的支路选择用于输出的数据。该随后的时钟是在第一个时钟之后的一些数目的时钟脉冲之后出现的一个时钟信号。

在一具体的实施例中,第一时钟信号之后的时钟数是1(即实现具有2次等待间隔的流水线效应)。在另一实施例中,第一时钟信号之后的时钟数是2(即实现3次等待间隔)。本发明的另一个实施例可实现在2和3之间的一个可编程选择等待间隔。

系统分支的每一个都包括一对锁存器,其一表示逻辑“1”输入信号,而另一个表示逻辑“0”输入信号。复位电路提供来用于复位每个流水线分支中的这一对门锁存器。在具体的实施例中,一旦有来自输出的该支路的消选信号,就立即将这一对锁存器的每一个复位。在每一个周期中有单一的支路被复位。

采用同步环绕式计数器系统与导流电路结合工作,在其中导流及锁存有数据的这些支路循环操作。该计数器还直接用于选择用以输出数据的支路。随每一个时钟脉冲,该计数器自动启动且递增很快。

还包括导流电路。对于等于一个周期时间的时段上,该导流电路顺序地择用多个信号之一,恰在一个窄数据脉冲到达之前,该顺序被开始计时递增。相对于时钟周期的时间

而言，数据脉冲最好是要窄。该导流信号的择用在窄数据脉冲到达以前操作，以把数据脉冲导流到若干个系统支路的一个中。比数据脉冲到达稍早一点，利用一个共用定时信号来实现该导流信号择用的精确相关定时。在一个具体的实施例中，该共用定时信号是通过在若干个预解码信号线之一上的脉冲的出现而被产生的。该共用定时信号再被用以选通读出放大器(从而产生出数据脉冲)并提供导流信号的精确定时。最近在系统的数据路径中的这种共用定时信号的使用使得数据信号能以高频率导流。这种设计方案能够适应由于温度、供电电压之类原因所引起的定时的变化。

还提供有以逐个周期为基础禁止系统输出的电路。本发明的其它的具体的实施例还具有这样的特征：即使在可以获得数据之前选择了用于输出的数据，也能防止流水线输出中的误操作。

本发明的结果是小型、高速的具有可编程等待间隔的流水线电路，利用对于产生数据和把信息导流入若干寄存器之一中属于共用的定时信号而实现精确定时。由于没有采用传送门，数据从存储器电路的输入非同步地流向流水线电路的输入。结合附图所作的充分描述将会有对本发明的性质及优点有更充分的了解。

根据本发明的一个方面，提供了一种具有带输出端的数据通路的集成电路，包括：时钟输入端，用于接收时钟信号；用于存储数据的具有多个分支的寄存器；具有连接到携带数据的数据线的输入端的导流电路，所述导流电路连接到所述寄存器，并且随后响应由所述时钟信号触发的第一信号，将

所述数据导流到所述寄存器的所述多个分支中的一个已选择的分支;和连接到所述寄存器的分支选择电路,所述分支选择电路随后响应由所述时钟信号触发的第二信号,将存储在所述多个分支中的已选分支中的数据送到输出端,其中,在第一数据开始通过数据通路传播之后,但在所述第一数据到达输出端之前,第二数据开始通过数据通路传播。

根据本发明的一个方面,提供了一种存储电路,包括:用于存储数据的可寻址存储单元的阵列;连接到存储单元的所述阵列的数据线;时钟输入端,用于接收时钟信号,所述时钟信号的第一沿启动第一读周期以将输出数据提供到所述数据线上;用于存储数据的具有多个分支的寄存器;产生多个导流计数状态和选择计数状态的计数器,所述多个导流计数状态和选择计数状态随着所述时钟信号的周期而变化;连接到所述寄存器和所述数据线的导流电路,所述导流电路进一步接收所述多个导流计数状态,所述导流电路将所述输出数据导流到所述寄存器的所述多个分支中的一个已选择的分支;和由所述多个选择计数状态驱动的选择电路,用于进行选择以输出存储在所述已选择分支中的所述输出数据,其中,所述导流电路响应由所述时钟信号的所述第一沿触发的第一导流计数状态,将第一输出数据导流到所述寄存器的第一分支中,并且所述选择电路响应由在所述时钟信号的所述第一沿后的多个周期出现的所述时钟信号触发的第一选择计数状态,选择所述第一输出数据以进行输出。

根据本发明的一个方面,提供了一种集成电路,包括:连

根据本发明的一个方面,提供了一种集成电路,包括:适于接收时钟信号的时钟输入端;响应所述时钟信号来产生第一、第二和第三导流计数状态和第一、第二和第三选择计数状态的计数器;用于存储数据的具有第一、第二和第三分支的寄存器;具有从数据线接收数据的第一输入端和分别从所述计数器接收导流计数状态的第一、第二和第三导流控制输入端的导流电路,所述导流电路连接到所述寄存器,所述导流电路响应来自所述计数器的所述导流计数状态来将数据导流到所述寄存器的所述第一、第二和第三分支中的一个已选分支中;和具有分别连接到所述寄存器的所述第一、第二和第三分支的输出端的第一、第二和第三数据输入端,和分别接收来自所述计数器的选择计数状态的第一、第二和第三选择控制输入端的分支选择电路,所述分支选择电路响应所述时钟信号来将存储在所述第一、第二和第三分支中的已选分支的数据送到输出节点,其中所述选择的分支是响应于所述选择控制信号来选择的。

根据本发明的一个方面,提供了一种对出自同步存储系统的数据进行流水线传送的方法,所述同步存储系统具有带有输出端的数据通路,用以存储数据并带有多个分支的寄存器,和接收时钟信号的时钟输入端,该方法包括步骤:在所述时钟输入端接收第一时钟沿,所述第一时钟沿启动第一数据的第一读周期;通过所述数据通路传播所述第一数据;将由所述第一读周期产生的所述第一数据导流到所述寄存器的所述多个分支的第一个分支中;在所述时钟输入端接收第二



时钟沿,所述第二时钟沿启动产生第二数据的第二读周期;在所述第一数据到达所述输出端前通过所述数据通路传播所述第二数据;和将由所述第二读周期产生的所述第二数据导流到所述寄存器的所述多个分支的第二个分支中。

根据本发明的一个方面,提供了一种对在具有带输出端的数据通路的电路中的数据进行流水线传送的方法,该方法包括步骤:沿所述数据通路开始传播第一数据;在开始传播所述第一数据且在所述第一数据到达输出端前,沿所述数据通路开始传播第二数据;将所述第一数据导流到第一存储元件;在导流所述第一数据步骤之后将所述第二数据导流到第二存储元件;选择在所述第一存储元件中的所述第一数据以进行输出;和在选择所述第一数据的步骤之后选择在所述第二存储元件中的所述第二数据。

根据本发明的一个方面,提供了一种在具有数据通路和输出端的存储电路中,一种对数据进行流水线传送的方法,包括步骤:开始由时钟信号的第一沿触发的第一读周期,所述第一读周期产生第一数据;通过所述数据通路传播所述第一数据;开始由在所述第一沿后的时钟信号的第二沿触发的第二读周期,所述第二读周期在所述第一数据到达所述输出端之前产生第二数据;通过所述数据通路传播所述第二数据;将所述第一数据导流到第一存储元件中;在导流所述第一数据的步骤后,将所述第二数据导流到第二存储元件中;选择在所述第一存储元件中的所述第一数据以进行输出,选择所述第一数据的步骤由所述时钟信号的所述第二沿触发;

和选择在所述第二存储元件中的所述第二数据以进行输出，选择所述第二数据的步骤由在所述第二沿后的所述时钟信号的所述第三沿触发。

### 附图说明

图 1 是根据本发明具体实施例的导流电路一部分的简化电路图；

图 2 是根据本发明具体实施例的导流电路的电路图；

图 3 是当采用等待间隔为 2 时图 2 电路的定时图；

图 4 是当采用等待间隔为 3 时图 2 电路的定时图；

图 5 是当等待间隔为 1 时的导流信号的电路图；

图 6 是用于把图 2 和 5 的输出传送到输出缓冲级的输出电路图；

图 7 是使用在图 2 电路中的、用于产生流水线计数的流水线计数产生电路；

图 8 是使用在图 2 电路中的、用于产生正时信号的流水线延时电路；

图 9 是计数延时产生电路，用于产生与本发明实施例之一结合使用的定时信号；

图 10 是使用在图 2 的复位电路部分中的复位信号产生电路；和

图 11 是与图 2 电路结合使用的流水线启动电路。

### 具体实施方式

参考图 1, 示出根据本发明的等待间隔导流电路 100。该电路是示于图 2 中的三个完全相同部分或支路之一。与图 2 的电路相比它有某种程度的简化, 但它包括有出自示于图 6、8 和 11 的其它电路的功能的获得整个操作的总体概念。导流电路 110 用以导流数字信号, 以等效的或真实的 I/O 信号线(后称为 I/O# 和 I/O) 到一个择用分支的分别的节点 N01 或 N04, 并从该节点到例如同步存储器装置的输出缓冲单元 150。该电路使用若干个控制信号, 以确保来自 I/O 线的的数据以精确的定时传播到输出缓冲单元 150。信号的导流可被用来从存储器装置快速而准确地流水线输出数据。本专业技术人员将会清楚, 本发明可与包括 DRAM、SRAM、ROM、快速存储器之类的任何同步存储装置结合使用。而且本发明的流水线方案可做修改的使用在其它需要高效、高速流水线的应用场合。

在说明书中, 将描述本发明的特定的实施例。尤其要讨论设计使用在 DRAM 系统的输出数据路径中的一个实施例。在该特定实施例中, 数据是以脉冲(例如 1—5ns 宽的脉冲)的形式传播通过 DRAM 的。在最佳实施例中, 其电路可用由题为“在 CMOS 集成电路中的快速传播技术”(流水号 08/269,451, 1994 年 6 月 30 日提交) 相关共同受让申请所描述的快速传播 CMOS 逻辑来实现, 该申请结合于此作为参考。本专业的技术人员会认识到, 也可使用其它的技术方案。

在该特定实施例中启动一个读周期的时钟周期将引发四个动作的出现。首先是将一个电路(例如图 1 所绘支路节

点N01和N04)的一个支路的两个存储节点被复位。随后:将与此同时地完成传统的非同步读出访问,以在I/O或I/O<sup>#</sup>线上的数据脉冲作为结束。第三,读周期将引起导流信号的产生,该信号将在稍先于非同步数据脉冲的到达而被择用。最后,导流信号将使数据以I/O或I/O<sup>#</sup>导流到一个合适的存储节点,将该存储节点置1。随后的时钟周期将选择这一数据用以输出。该数据可被输出到任何数目的接收电路(例如,处理器、高速缓冲存储器等等)。

为实现一个为2的等待间隔,至少需要两个电路100,而至少要有三个并联存在的电路100来实现为3的等待间隔。通过提供更多的电路100就可以得到更多的等待间隔。在一特定实施例中,本发明的流水线电路将使得用户确定从1到3个等待间隔的具体要求。图1的等待间隔导流电路100的几个确定的方面已被总转化,以简化电路操作的解说。更详细的内容将结合后序的附图给出。

在一具体实施例中,来自存储器装置的I/O信号在等待期间为高值。所以,在任意一条指示有数据存在的线上是低值信号。这些信号通常是负数据脉冲,例如可以是2ns宽的脉冲。在一具体的实施例中,I/O信号是在随着存储器读请求的一个大约10ns的非同步延时之后而被产生来的。可是,由于该流水线传送的使用,存储器请求可以小于该存取时间的几个间隔作出。为2的等待间隔将流水线传送数据,以使I/O信号每7.5ns出现一次,并在输出缓冲单元上每7.5ns替换一次新数据,为3的等待间隔将使I/O信号及新输

出数据以 5ns 的间隔出现。

PMOS 晶体管(例如 102 和 104)的符号包括一个圆圈(小泡)在其栅极。NMOS 晶体管(如 114 和 116)的符号就没有此小圆圈。在 PMOS 晶体管 102 和 104 源极上方的短水平线代表正电源  $V_{cc}$ 。NMOS 晶体管 114 和 116 的源极接地。

当 I/O# 或 I/O 线脉动到低电平,PMOS 晶体管 102 或 104 即被接通。与此同时,PIPECNTDELO 被择用为低值,在一电路 100 中被启动来接收数据,(在其它电路 100 中,PIPECNTDEL1 和 PIPECNTDEL2 是高电平)。在 I/O# 或 I/O 上的低电平脉冲与在 PIPECNTDELO 上的低电平相结合,使事先复位或低电平的节点 N01 或 N04 被设置,即移到正电源电压  $V_{cc}$ 。节点 N01 或 N04 实际包含一对数据锁存器。节点 N01 由所收逻辑“1”置为高值,而节点 N04 由所收逻辑“0”置为高值。

PIPECNTDELO 是由将结合图 7.8 和 9 详细讨论的电路和 110 所产生的信号。PIPECNTDELO 的择用定时受电路 110 控制。一般说来,PIPECNTDEL 信号被定时成与在 I/O 线上信号的到达相结合地现出现。属于相对长持续期的 PIPECNTDEL 信号产生出一窗口,其中可能出现 I/O 信号。响应信号(例如 PIPECNTDEL1 和 2)送到此图没示出的电路分支。启动对 PIPECNTDEL(1、2 或 3)的哪一个的选择,并依据所选择的等待间隔(例如,2、3 或更多)及计数器(PIPECNT)的状态来接收从一周到另一周期的数据改变。这种选择受电路 112 控制。电路 110 和 112 将在后结合图 8

作详细讨论。电路 110 和 112 共同将负数据脉冲从 I/O# 或 I/O 线导流到被启动接收数据的电路 110 的节点 N01 和 N04。“导流”地含意是,当信号 PIPECNTDEL0 择用为低值,如果信号(例如负脉冲形式的信号)出现在线 I/O# 上,则将启动支路 0 以接收数据,信号将被导流到节点 N01,设置该节点为高值而不影响在其它电路 100 中的对应节点。如果负脉冲到达 I/O 线,且如果是信号 PIPECNTEDL0 被择用为低值,则数据被导流到节点 N04,设置该节点为高电平。如果信号达到线 I/O 或 I/O# 且 PIPECNTDL0 不被择用低电平(因为 PIPECNTDEL1 或 2 被择用低电平),该数据则被导流到另一个支路且节点 N01 和 N04 都不改变。这种特征使得以连续的周期抵达 I/O 线的单一设置上的信号被导流到若干不同支路中的不同节点,以实现流水线传送效应。

一旦数据被正确地导流到一特定节点(例如图 1 的节点 N01 和 N04),就在那里等待,直到有正确的时钟输入到来。一旦有正确的时钟输入的到达,PCNT0 被快速选为高电平。这使得数据随即被传播到一个输出缓冲单元 150。PCNT0 信号实际上是从一个 PIPECNT0 信号中提取的,该信号是由将结合附图 7 所讨论的一个计数器产生的,只有在计数信号 PIPECNT0 已被选取高值、且输出将在本周期被启动,PCNT0 信号才被选为高值。在一个具体的实施例中,输出控制电路包括有使得输出被关断(开路)的特征,从而空闲存储器输出缓冲单元,以承接其它信号。这种控制电路将结合图 6 和 11 作深入的详细讨论。

PCNT0 的选择接通 NMOS 晶体管 126、134 和 140。即与节点 N01 或 N04 上的一个高“设置”电平结合,使输出缓冲单元 150 被移到高或低电平。例如,已经有负脉冲达到线 I/O 上,当 PIPECNTODEL0 被择用为低值,则晶体管 104 和 108 将同时接通以设置节点 N04 为高值(N01 和 N04 都事先被复位成低值)。节点 N04 耦合到 NMOS 晶体管 128 的栅极。随着节点 N04 变高,晶体管 128 被接通。时钟输入的接收(该输入之后则起始此读周期)将引出 PIPECNT0 和 PCNT0(输出启动)的选择,它将接通 NMOS 晶体管 126。随着晶体管 128 和 126 都接通,PMOS 晶体管 132 和 142 的栅极都被接地,将这些晶体管接通。当 PIPECNT0 被选为高值时,PIPECNT1 则不被选择,即为低值。如将被更详细讨论的那样,这些 PIPECNT 信号由一个高速同步环绕式计数器所产生。在任何给定的时间上,只有这些信号(PIPECNT0—2)之一被选择成高值。利用 PIPECNT1 的低值,PMOS 晶体管 130 和 144 被接通。由于晶体管 130 和 144 都导通,节点 PULLND 被移到  $V_{\infty}$ ,导通 NMOS 输出晶体管 148,随着 PMOS 晶体管 144 和 142 都导通,节点 PULLUP 也被移到  $V_{\infty}$ ,关断 PMOS 输出晶体管 146。因而使输出缓冲单元 150 被接到地电平。此时节点 N01 仍复位成低电平,使 NMOS 晶体管 136 和 138 截止。以相似方式,如果数据已经到达线 I/O#,带点 N01 已通过 PMOS 晶体管 102、106 被置成高电平,则节点 N04 仍然被复位成低电平。在此情形中,当 PCNT0 被选择成高电平时,则线 PULLUP 和 PULLDN 将经过晶体

管 140、138 和 134、136 被移到地电平。这将使 PMOS 晶体管 146 导通而 NMOS 晶体管 148 截止,把输出缓冲单元 150 移到高电平。

由于信号被传播经过该系统的高速度,线 PULLUP 和线 PULLUP 潜在有输出误操作的危险。有几种用来避免输出中误操作的措施。首先,一旦当前 PIPECNT(输出)周期结束,作为下一个时钟输入的结果,PMOS 晶体管 130 和 144 将被变成非导通态,因为下一个管道计数 PIPECNT1 被很快选成高值。且在此时 NMOS 晶体管 126、134 和 140 通过 PCNT 的低值的消选而被截止。由于 NMOS 晶体管 134、140 和 PMOS 晶体管 130、144 全部截止,该特定支路将不再受 PULLUP 或 PULLDN 状态的影响。结果是,PULLUP 和 PULLDN 保持其目前电压,直到由下一个所选支路(或由输出禁止电路)驱至新电压为止。当同时出现当前支路的消选及下一支路的选择的情况,如果数据尚未达到下一支路的节点 N01 或 N04 的等效节点上(例如支路 1 的节点 NO2 和 NO5),仍不会在输出中有误操作。反之,伴随下一个分支的被选择将什么也不发生。只有当数据最终达到下一支路的节点 NO2 或 NO5,节点 PULLUP 或 PULLDN 才发生电压的改变。所以,输出被保持在高、低或开路状态,直到可作为下一输出的数据能得到为止。如果该下一数据恰好是与当前数据相同,则节点 PULLUP 和 PULLDN 将不改变电压并因此不会有输出的误操作,即使是在该数据之前下一个 PIPECOUNT 选择数据是可获得的情况下仍然如此。



本专业的技术人员明白,PULLUP 的上升可被设计得比 PULLDN 的上升更快,使得 PMOS 晶体管 146 的截止比 NMOS 晶体管 148 的导通更快。同样,PULLDN 的下降可被设计得比 PULLUP 的下降更快,使得 NMOS 晶体管 148 的截止比 PMOS 晶体管 146 的导通更快。这将会使能耗减小。

为保证将有效信号置于节点 N01 和 N04,提供有 RESET0 信号。在一具体实施例中,对于支路 0 而言(由 PIPECNT0 选择来输出数据的支路),该信号被选来在一周期的开始作为正脉冲,在该周期中,PIPECNT1 已被选为高电平(即紧靠在选择了 PIPCNT0 以输出数据的周期之后)。RESET0 信号被耦合到 NMOS 晶体管 114 和 116 的栅极。具有正脉冲的信号的择用将节点 N01 和 N04 复位成地电平,这是表示在这两节点中尚无数据得到的条件。信号还由非门 120 反相,其输出导通 PMOS 晶体管 124,它复位 PMOS 晶体管 132 和 142 栅极的高电平,这又是表示没有可得到的数据的条件。一旦这些节点的每一个都被正确复位,该电路则被准备好在下次 PIPECNT0EL0 再次将数据导流到支路 0 时接收在 I/O 或 I/O# 上的负脉冲形式的新数据。在下次 PIPECNT0 被选为高电平时,新数据将被输出。应注意,在由 PIPECNT0EL0 将新数据经 I/O 和 I/O# 导流进入该具体支路(支路 0)之前,复位脉冲定时必须结束。在本发明的一个具体实施例中,当把等待间隔设置为 3 时,有一个 10ns 的窗口,在该窗口中,在新数据达到节点之前,RESET 的开始和结束均已完成。此时间与时钟周期时间无关。在等待间隔为

2 的情况下,有更多的时间可用。本专业人员有能力在该窗口内调节 RESET 的定时。

虽然电路 100 在某种程度上简化,但本发明的几个方面都被示出。首先,除非在 I/O 和 I/O# 信号出现之时有 PIPECNTDEL0 信号被择用为低值,出现在 I/O 线上的信号将从被传播(导流)到节点 N01 或 N04。由于 I/O 信号可以是以快速间隔(例如在某一实施例中为 5ns)出现的例如为 2ns 宽的脉冲,所以需要 PIPECNTDEL0 信号的正确定时。这一定时是由延时电路 110 来协调的。在任何给定周期内的 PIPECNTDEL 信号被择用的选择是由在读请求被发出之时的 PIPECNT 计数器的状态所决定及由选择的等待间隔所决定。这是由图 8 的电路 440 完成的。一旦数据将节点 N01 和 N04 设置成高电平,只有当 PCNT0 被选择时该数据才被选择作为输出;反过来,PCNT0 的选择只有当 PIPECNT0 被选择时才会出现(且由输出启动电路 118 启动数据的输出)。就是说,直到迟于若干时钟之后而已经选择了 PCNT0 信号,出自节点 N01 或 N04 的数据将不会被置于输出缓冲单元 150。此种门控效应被用来控制数据的流水线传送。通过复制图 1 的电路 100N 次,能够形成具有从 2 至 N 的任何深度的数据流水线。

参考图 2,其中示出根据本发明的一个导流电路 200 的一个具体实施例。该电路具有三个相同支路,实现为 2 或 3 的等待间隔。为 1 的等待间隔由稍后讨论的单独的电路实现。该三个支路的每一个的主要部分都是图 1 所示电路 100

的复制。本专业的技术人员明白,可以通过使用更多数量的支路达到更多的等待间隔。图 1 中的某些特征,例如延时电路 110、等待间隔选择电路 112、输出控制电路 118 及输出晶体管 146 和 148 在图 2 中没有示出。相反,这些特征的实施例的具体情况将作深入讨论。

本发明一个具体实施例的几个不包含在图 1 中的特征被示于图 2 中。具体地说,存储节点 N01—N06 的每一个都包括一个由一对非门形成的锁存器。这些锁存器是用来在即使当系统操作在担心会有泄漏电流的低频条件下确保信息被保持在这些节点的每一个中。它们的构成具有足够高的阻抗以保证它们能承受它们分别的复位脉冲或在 I/O 或 I/O<sup>#</sup> 上的新数据所加的过载,它们还具有足够低的阻抗以克服在它们分别节点上的任何泄漏电流。贯穿本说明书中的类似的锁存器也都被它们分别的输入脉冲所过载的供电。通过在这些节点上提供锁存器,实现了以非常低的频率的操作。附加的 PMOS 锁存器 264A—C 被设置来保持节点 N013—N015 为高电平。一旦这些节点被置成高值,锁存器 264A—C 将保持该高电平直到出现一个信号的改变为止。它选择了它们各自的 RESET 信号时,节点 N013—N015 由 PMOS 晶体管 224A—C 复位为高电平。应当注意,该 RESET 功能并不复位在线 PULLUP 或 PULLDN 上的信号,相反,这些信号被保持到由新的输出数据所取代为止。这样有助于保证无误操作的输出。例如,在第一周期中假设在晶体管 202A 栅极上收到来自 I/O<sup>#</sup> 的负数据脉冲,并由晶体管 206A 导流到支路 0 的

存储节点 A,使得当 PCNT0 被选成高电平时 PULLUP 和 PULLDN 被拉到地电平。在下一个周期下,如果数据再次出现在线 I/O# 上,它将被导流到支路 1 的节点 NO2 上,将其设置为高电平。这将使晶体管 236B、238B 将线 PULLUP 和 PULLDN 拉到地电平(一旦选择了信号 PCNT1)。可是该线已经在地电平,所以最终的数据输出保持在高电平。即使是在数据到达以设置节点 NO2 为高电平之前选择了 PCNT1,也仍是此情况。

当 PULLUP 和 PULLDN 需要被拉到高电平以在一输出周期从支路 1 输出数据时(即数据已经由真实的 I/O 线进入并被导流以设置节点 N05 为高电平),一旦选择了 PIPECNT1 和 PCNT1,节点 N014 就被拉到地电平。这使得 PULLUP 和 PULLDN 被接到高电平,因为 PMOS 晶体管 230B 和 244B 是导通的,它们的导通是由于无论何时 PIPECNT1 被选为高电平,都会使 PIPECNT2 被消选低电平。而且,每一个 PIPECNT 周期的结束都是通过截止先前选择的 PMOS 晶体管 230、244 和先前选择的 NMOS 晶体管 226、224 而对线 PULLUP 和 PULLDN 进行隔离而完成的。因为(例如)PIPECNT1 被消选低电平而同时 PIPECNT2 被选成高电平,对支路 1 进行消选,所以这些晶体管被同时截止。由于 PULLUP 和 PULLDN 在每一个周期的结束与一条支路相隔离,这些节点能立即接受新数据,作为所提供的下一条支路。

应注意到,即使在新数据到达之前对其请求,该电路也能逐个周期地提供无误操作的(但被延时的)过渡。再次假定

从读周期的起始到在 I/O 或 I/O<sup>#</sup> 上的负脉冲的产生的非同步延时是 10ns。在本例中的数据地导流到支路 0。还假定是一个为 2 的读等待间隔,即,一旦收到紧随被启动的所关心的读周期的时钟的第一个时钟信号,即选择用以输出的数据。如果时钟周期的时间大于 10ns,则在其被选作输出之前,数据将已经非同步地传播通过了存储器,就是说,在 PCNT0 被选成高电平前,节点 N01 或 N04 将已被设置为高电平。如前所述,紧接着下一个外部时钟的接收,PCNT0(对于后续周期是 PCNT1 和 PCNT2)被选择为高电平且有新数据出现在输出上。另一方面,如果周期时间是小于 10ns,则在其可获得之前就对数据请求。即在节点 N01 或 N04 已被设置为高电平之前 PIPECNT0 就被选成高电平,且这两个节点仍处于它们复位的低电平状态。在此情形中,直到非同步传播的数据到达之前,由下一个时钟对于 PIPECNT0 的选择将不引起对输出的改变。就是说,节点 PULLUP 和 PULLDN 保持其原来状态且其输出也保持其原来的高电平、低电平或开路状态,直到新数据到来为止。当 I/O 或 I/O<sup>#</sup> 负数据脉冲最终出现时,节点 N04 或 N01 被置为高电平,且新数据立即传播到输出,PIPECNT0 仍被选择成高电平。而且,如果新数据与旧数据完全一样,则输出保持恒定而无误操作。而且如若输出切换,它将保留先前的数据为有效,直到可获新数据取代它为止。输出的状态不改变,直到下两个条件被满足为止:根据已经非同步地传播通过存储器,以在输出端可以获得;以及已经收到了合适的时钟以选择该数据作为输出。这

两个条件哪一个先出现无关紧要。但是,如果非同步传播经过存储器的数据在当 PIPECNT 先于下一个试图选择该数据作为输出的时钟时仍不能被获得,就会增加源于时钟的存取时间。

现在来描述以不同等待间隔操作的电路 200。再次假设数据非同步地传播经过存储器将用 10ns,且由此设置或是节点 N01 或 N04 为高电平。还假设在例如节点 N01 或 N04 被置为高电平且 PIPECNT 被选成高电平之后,要 5ns 使输出变为有效。参考图 3,其中示出了为 2 的等待间隔的定时图。在本例中,大约每 11ns 作出一个数据请求。图 3 示出了把输出信号传播通过图 2 电路 200 的单一支路所需要的信号间的相互作用。具体而言,数据被示为由 PIPECNTDEL0 所导流,以便将节点 N01 或 N04 置为高电平(根据是否有负的数据脉冲出现在 I/O<sup>#</sup> 或 I/O 上)。由 PIPECNT0 和 PCNT0 选择的该支路被称之为支路 0。箭头表示这些信号间的关系(例如一个信号引起第二信号转换的上升和下降沿)。象参考图 11 而变得显而易见的那样,该 PCNT 信号通常是等效于分别的 PIPECNT 信号。只有当输出已被禁止时它们才有差异。因此,为了进行讨论的目的,PIPECNT 信号将被参考。

如图 3 所示,在任何时间只选定一个信号 PIPECNT0—2 (高)。每个时钟周期的上升沿触发信号间的跃变。例如,如果 PIPECNT0 为高电平,时钟的下一个上升沿将使 PIPECNT1 变为高电平,使 PIPECNT0 变为低电平,时钟信号的上升沿也驱动 RESET 信号。在一具体实施例中,RESET0

信号被 PIPECNT1 的上升沿触发,使支路 0 节点 N01 和 N04 复位到低电平,N013 到高电平。同样,PIPECNT2 的上升沿导致 RESET1 信号的产生,而 PIPECNT0 的上升沿引起 RESET2 信号的产生。这些信号的产生和定时将结合图 10 讨论。

在一特定时钟脉冲,为方便起见标为时钟脉冲 0,进一步标为起始读取时钟脉冲,产生存储器读取请求。从请求时间开始,在近似 10ns 的非同步传播延时之后,该时钟脉冲导致从存储器输出一信号作为 I/O 或 I/O<sup>#</sup>上的负脉冲。在图 3 所示的实例中,启动读取周期碰巧在引起 PIPECNT2 的上升沿的时钟周期发生。在读取周期开始后的一段时间,与其后是否接收任何附加时钟脉冲无关,使 PIPECNTDEL0 维持低电平(如将描述的,选择 PIPECNTDEL0 而不选择 PIPECNTDEL1 是选择等待间隔为 2 的结果)。对在一具体实例的实例中,在时钟脉冲 0 的开始读周期之后 PIPECNTDEL0 维持低电平 8.5ns。因而,与时钟频率无关,此数据被导流到支路 0;也就是将节点 N01 或 N04 都置位于高电平(根据该信号是否出现在 I/O<sup>#</sup>或 I/O)。在 PIPECNT2 信号选定之后,具体地讲,刚好在数据脉冲到达 I/O 或 I/O<sup>#</sup>之前,使 PIPECNTDEL0 信号维持一特定时间。提供充分的裕度,以确保 I/O 信号在由电流 PIPECNTDEL 信号(此处是 PIPECNTDEL0)限定的窗口内到达。此裕度允许由温度、电源、或其它因素引起的信号定时的变化。PIPECNTDEL 信号的产生和定时将在下文结合图 8 做更详细地描述。

因为在维持 PIPECNTDEL0 信号低电平的过程中,出现了与读取周期相关的负 I/O 数据脉冲,则将数据导流到支路 0 中的一节点(即节点 N01 或 N04)。负脉冲的控制把节点 N01 或 N04 从其复位的低电平置位到高电平。该节点在被后来的 RESET0 脉冲之前(即在完成读出该数据的周期之后)保持高电平(在一实施例中由于锁存)。当收到下一个外时钟输入,时钟脉冲 1 时,使下一个流水计数 PIPECNT0 尽快地选定高电平,而使前一个流水计数 PIPECNT1 尽快地选定低电平。这就选择了新的支路,而脱离了前一个选定的支路。当 PIPECNT0 被选为高电平时,晶体管 234A、226A 和 240A 导通,并且如果节点 N04 置位于高电平,则节点 PULLUP 和 PULLDN 被拉到高电平,如果节点 N01 置位于高电平,则被拉到低电平。然后,将这些信号传播到存储器件的输出缓冲级 150。

到达 I/O 线的下一个数据将被导流到支路 1 的节点 N02 或 N05,那是因为计数器已被递增,而 PIPECNTDEL1 信号将维持低电平。当在时钟脉冲 2 后立刻选定 PIPECNT1 时,此数据将驱动线 PULLUP 和 PULLDN。一周期后,因信号 PIPECNTDEL2 维持为负,而将在 I/O 线上收到的数据导流到支路 2 的节点 N03 或 N06。当收到时钟脉冲 3 选定信号 PIPECNT2 时,线 PULLUP 和 PULLDN 将再次收到新数据。这一过程将连续将数据导流到支路 0、1、2,0、1、2、0 等等,并从支路 0、1、2,0、1、2、0 检索数据。第一数据可根据读取周期起始所存在的计数被导流任一支路。在完成最后读取周期



后,可使输出开路。为完成此操作的电路将在下文结合图 5、6 和 11 做描述。

图 3 的定时图表示在选定输出之前,即超过 10ns 的时间周期,到达 I/O 或 I/O<sup>#</sup>上的数据。但,如后文所述,也可选取 7.5ns 的时间周期。

现在参照图 4,表示等待间隔为 3 的实例同步图。对等待间隔为 3,在将允许高频运行的读取周期之后,将使数据输出两个时钟脉冲。该接线图再参照图 2 的电路 200。也可使用同样的电路 200,通过产生由 PIPECNT 信号所维持的那个 PIPECNTDEL 信号的不同选择,获得为 3 的等待间隔。

图 4 表示在等待间隔为 3 和大约 5.5ns 的周期时间的运行。图 4 中时钟信号的上升沿引起 PIPECNT 信号间的转换。再有,定时图表示数据被导流到支路 0 的节点 N01 或 N04(即在 PIPECNTDEL0 信号维持低电平时,在 I/O 线上出现负脉冲)。在 I/O<sup>#</sup>或 I/O 上的低脉冲和在 PIPECNTDEL0 上的低电平信号的结合分别将数据导流到置位于高电平的节点 N01 和 N04。PIPECNT0 信号的选择引起数据向节点 PULLUP 和 PULLDN 的传递,作为输出连续出现在存储部件的缓冲级 150 上。但,对于等待间隔为 3,在开始读取周期的时钟脉冲之后的第二个时钟脉冲,即时钟脉冲 2 才出现 PIPECNT0。应注意,在选定的等待间隔为 3 时,在 PIPECNT1 选定高电平之后,PIPECNTDEL0 维持低电平一段固定时间,然而,对等待间隔为 2 在 PIPECNT2 选定高电平之后,PIPECNTDEL0 被维持在低电平相同的固定时间。在一具有

10ns 非同步延时的具体实施例中,在 PIPECNT1 选定之后 PIPECNTDELO 信号将被维持低电平 8.5ns(同时 PIPECNT2 不择用高电平)。

图 2 的导流电路 200 的运行允许很快地并有效地以流水线输出数据,支持高频运行。为特定系统适当选择等待间隔允许用户或存储器的设计者来优选存储器的效能。

当等待间隔为 2 时,在读取周期开始之后的第二时钟脉冲从电路 200 输出数据。即,在时钟脉冲 0 启动读取周期,在时钟脉冲 1 从该读取周期选择待输出的数据,并在时钟脉冲 2 将有效输出数据锁存到接收电路中。可以使用电路 200,将数据输出给任意数目的接收电路。比如,可把数据供给超高速缓冲存储器系统或中央处理单元。对一个从读取周期开始至数据到达节点 N01 或 N04 之间的非同步延时为 10ns 通过输出缓冲器的非同步延时为 5ns 的实施例中,在读取周期起始后的 15ns 之内数据不能变为有效的输出。即,从时钟脉冲 0 至时钟脉冲 2 之间的时间必然是至少有 15ns 用于被接收电路所接收的校正数据。在等待间隔为 2 时,周期时间必然大于或等于 7.5ns。应予注意,当在 7.5ns 周期时间时,在读取周期开始后在该数据将节点 N01 或 N04 置位为高电平之前的 2.5ns,称 PIPECNT0 为新的数据一时钟脉冲(或 7.5ns)。在时钟脉冲 1 选定此输出数据后的 2.5ns,当数据最终到达节点 N01 或 N04 时,输出没有误操作,而仅仅替代开始切换。在输出近似 5ns 后时钟脉冲 2 使数据变为有效。

另一方面,对于等待间隔为 3,在时钟脉冲 3 将数据锁

存到接收芯片内。再有,对等待间隔为2,从读取周期开始非同步延时10ns后,数据将节点N01或N04置位为高电平,并在5ns内通过输出缓冲器。但在等待间隔为3时,在此15ns时间间隔内有三个时钟脉冲。在此等待间隔为3,在读取周期开始后的10ns,在时钟脉冲2以5ns周期间时为输出选定数据。当从时钟脉冲0非同步延时10ns后将N01或N04置位为高电平时,同样在时钟脉冲0之后的10ns由时钟脉冲2选定PIPECNT0,在输出5ns(即在15ns)后使数有效。这是在第三时钟脉冲同时发生的,将数据锁存到接收电路(即CPU、超高速缓冲存储器等等)。

在等待间隔为2和等待间隔为3的运行之间的主要作用的差别在于跟随所选定的PIPECNT的那个所维持的PIPECNTDEL的逻辑选择。选择确定在为输出选定数据之前将发生的时钟脉冲次数。另一差别在于控制有能力输出的周期。

当时钟频率增加时,最小可用等待间隔也增加。对上面的实例,在5ns周期时间运行要求等待间隔(至少)为3。在7.5ns周期时间运行要求等待间隔(至少)为2。在等待间隔为1的运行(使用现在描述的电路)会看到对至少为15ns的周期时间是可行的。为使存取时间降至最短,系统的设计者应选取以其目标运行频率工作的最短的等待间隔。本发明的有效可编程序的等待间隔允许等待间隔的选取,因而使设计者能优选实际需要的系统。在不综合考虑其它的存储器设计参数如速度、功率和衬底面积的利益的情况可获得这种伸缩

性。

在一具体的实施中,用一所设计的附加电路来扩大导流电路 200,以支持等待间隔为 1。现在参照表示电路 280 的图 5,当期望等待间隔为 1 时可以使用。将这些电路一起用于单一存储器部件,使用户能选拔甚至较宽的等待间隔范围。与电路 200 类似,电路 280 驱动信号 PULLUP 和 PULLDN 的状态。当系统(或设计者)选择使用等待间隔为 1 而不是 2 或更高时,可以使用电路 280。如果预期的等待间隔大于 1,驱动信号 CL1(列读取等待间隔=1)为低电平,而非门 284 为门电平,或非门 286 和 288 的输出为低电平,而非门 290 的输出为高电平。所有四个晶体管 292、294、296 和 298 全截止。因而不能使电路 280 用于大于 1 的等待间隔。

但是,如果预期等待间隔为 1,图 2 的电路 200 不能启动,使图 5 的电路 280 启动。电路 280 的运行,当被启动时,依赖于输入信号 OUTOFF 的状态。如果 OUTOFF 为逻辑 1,非门 282 输出为 0,与非门 284 输出为 1,而或非门 286,288 输出为 0。晶体管 292、294、296 和 298 全截止。这允许图 6 的电路 300 使输出开路,如所描述的。如果 OUTOFF 为逻辑 0 而 CL1 为逻辑 1,则与非门 284 为低电平,启动电路为每个或非门 286、288 提供一低输入。还有,数据非同步地流过该存储器,在 I/O 或 I/O<sup>#</sup> 提供低脉冲。如果在线 I/O 上传播低脉冲,或非门 286 将输出高脉冲,而非门 290 将输出低脉冲,打开 PMOS 晶体管 292、294。此低脉冲有足够长的作用时间,将 PULLUP 和 PULLDN 拉到 Vcc。在下一周期切换它们

之前,它们将一直保持在  $V_{cc}$ (由于包括在图 6 的电路 300 中的锁存器)。因 PULLUP 和 PULLDN 都在  $V_{cc}$ ,驱动输出接地。同样,在线 I/O<sup>#</sup>的低脉冲引起由或非门 288 的高脉冲输出,因而使 NMOS 晶体管 296、298 导通。将 PULLUP 和 PULLD 全拉到地电位,驱动输出到  $V_{cc}$ 。直至被后一个周期切换为高电平,PULLUP 和 PULLDN 将保持接地。

应予注意,在等待间隔为 1 时,没有流水线传送。即,一旦在 I/O 或 I/O<sup>#</sup> 出现负脉冲形式,该数据依旧非同步地送到输出端。在切换输出之前不等待第 2 或第 3 时钟脉冲的到来。

使用信号 PULLUP 和 PULLDN 来驱动可以耦合到存储器件输出缓冲级的输出信号 OUT \_\_TTL。在一具体实施例中,由图 6 的输出电路 300 产生 OUT \_\_TTL。此电路 300 可以接受来自图 5 的电路 280(对于等待间隔为 1)和来自图 2 的电路 200(对于等待间隔大于 1)的输出。当用电路 200 或 280 将 PULLUP 和 PULLDN 驱动到  $V_{cc}$  时,PMOS 晶体管 314 截止,而 NMOS 晶体管 318 导通,因而将输出拉到地电位。当用电路 200 和 280 将 PULLUP 和 PULLDN 拉到地电位时,PMOS 晶体管 314 导通,而 NMOS 晶体管 318 截止,因而把输出驱动到  $V_{cc}$ 。为了快的驱动输出缓冲级,PMOS314 和 NMOS318 具有比较大的信道宽度。比如,在一具体实施例中,PMOS314 的信道宽度为  $400\mu$ ,而 NMOS 的信道宽度为  $250\mu$ 。

当 OUTOFF 维持高电平时,非门 302 为低电平,使

PMOS 晶体管 304 导通, 将 PULLUP 拉到 Vcc, 使 PMOS314 截止。在同一时间, NMOS316 导通, 将 PULLDN 拉到地电位, 使 NMOS 晶体管 318 截止。当两个晶体管 314 和 318 都截止时, 输出为开路。非门 306 和 308 锁住 PULLUP 的状态, 同时非门 310 和 312 锁住 PULLDN 的状态, 无限期地保持任何给定的状态, 直至新数据到达。信号 OUTOFF 实质上起输出截止命令的作用, 足以使 PMOS314 和 NMOS318 截止。

因而, 在选定的等待间隔等于 1 的情况下, 一旦在 I/O 线上产生一个信号, 不必等待任何附加时钟脉冲, 立刻放到输出缓冲级 (OUT\_TTL)。即, 使数据非同步地一直流到输出。当所选定的等待间隔等于 2 时, 从时钟脉冲 0 起始只有在接到下一个时钟脉冲 (即时钟脉冲 1) 之后才使来自读取周期的输出信号放到输出缓冲级上。当所选定的等待间隔等于 3 时, 从时钟脉冲 0 起始只有在接到时钟脉冲 2 之后才使来自读取周期的输出信号放到输出缓冲级上。其结果是能为比如来自同步存储器件的输出有选择的确定等待间隔。由于选择高的等待间隔系统能支持很高的运行频率。从读取周期至输出的存取时间是快的, 那是因为, 使数据从输入非同步地一直流至由 PIPECNTDEL 信号提供的导流。通过一系列中间传送门不使信号减慢。反之, 让信号通过一个单 (比如 PMOS) 导流门。不但电路比较简单, 而且不占什么衬底面积。当与同步 DRAM, 比如本发明的系统一起使用时, 用一块与其它的相似的非同步 DRAM 几乎同样的芯片可以提供输出流水线功能。

现在参考图 7,表示出 PIPECNT 产生电路 330。使用 PIPECNT 产生电路 330,为图 2 的导流电路 200 产生流水计数。概括地讲,PIPECNT 电路 330 是个高性能三驱动环绕同步计数器。在情况 3,有三种计数模式供选择,来支持具体的最大的等待间隔要求。

电路 330 包括一个可在一具体实施例中供 DRAM 系统使用的 LRAS 输入,如果 RAS 不是有效时,则使电路被禁止。当不读取是可能的话,在无效的 RAS 周期中,其特点是节省电能。线 LRAS 上的低信号使 NMOS 晶体管 346 截止,并通过 PMOS 晶体管 352 将节点 N20 拉到高电平。这就防止计数器递进计数,而将它锁定在它的现有计数。然而,当 RAS 为有效时,在 LRAS 上的高信号通过使 NMOS 晶体管 346 导通来启动电路 330,使计数被每个新的时钟脉冲递进。在 RAS 变为无效后,LRAS 信号必须在足够多的周期内保持有效,以便输出存储在电路 200 各支路中的数据。

当使用等待间隔为 1 时,无须计数电路 330。于是,通过将 CL1 信号导流为高电平可以禁止该电路。这就导致或非门 334 输出一个低信号。因而将逻辑 0 送到每个与非门 368、362 和 358 的一个输入端。这确保了节点 N15—N17 均保持低电平,而 NMOS 晶体管 372、366 和 360 全截止。在同一时刻,或非门 334 的低输出导致与非门 378、396 和 414 的高输出,它们的反相输出使 PMOS 晶体管 374、392 和 410 导通。于是,节点 N21、N22 和 N23 为高电平,使 PMOS 晶体管 384、402 和 420 截止。反相器 338 的输出为 NMOS 晶体管

340、342 和 343 提供高输入。使 PIPECNT0、1 和 2 全保持在地电位,以节省功率。

再有,如果存储器件仍没有衬底偏置(即,仍未将功率上升信号导流为高电平),则使电路 330 禁止。这是由发送一个通过非门 332 再到或非门 334 的 pwrup 信号,正如对等待间隔为 1 所描述的那样禁止电路 330 而达到的。本领域的技术人员将认识到,如果电路不仅在有衬底偏置(如在 DRAM 中)易于运行,而且在衬底偏置产生之前也允许运行的话,它会发生锁定。

流水计数电路 330 实质上是个自启动的从 Clkt2 到 PIPECNT0、1 或 2 的转换延时最小的同步计数器。Clkt 2 是个在时钟输入的上升沿之后快速发生的窄脉宽的正脉冲。在下面,将假定 LRAS 被导流至高电平,CL1 为低电平,而 pwrup 为高电平。即,使电路 330 启动。当在 Clkt2 上有正脉冲时发生计数。在 Clkt2 的上升沿,NMOS 晶体管 344 导通,将节点 N20 拉到地电位。这时,节点 N15、N16 或 N17 为高电平。为了解释清楚,假定在 Clkt2 出现时节点 N15 为高电平。如果在接收 Clkt2(PIPECNT0 和 1 为低电平)之前选定 PIPECNT2 为高电平,就应是这种情况。在 NMOS 晶体管的栅极的逻辑导致该晶体管导通,使节点 N21 快速跟随节点 N20 接地。如将描述的,在此时 PMOS 晶体管 374 为截止态。由反相器 380 和 382 形成的锁存器确保节点 N21 稳定在两种状态之一,直至迫使它成为相反的状态。将该锁存器设计得容易过负荷。节点 N21 的低电平使很大的 PMOS 晶体管



384 导通,以使信号线 PIPECNT0 快速选择高电平。由反相器 386、388 所形成的另一锁存器保持 PIPECNT0 上的高电平信号。当 PIPECNT0 变为高电平时,通过大 NMOS 晶体管 426 使先前选一的线 PIPECNT2 消选低电平。这确保从比如 2 的计数到 0 的很快的转换。

PIPECNT0 上的高电平信号导致反相器 388 的低电平输出,引起与非门 378 的高输出,引起非门的低输出。经这些逻辑门的很少的延时之后,使 PMOS374 导通。这发生在 Clkt2 上的脉冲的下降沿之后。其结果,PMOS 晶体管 374 使节点 N21 复位为高电平,但仅在 PIPECNT0 切换为高电平后才使 PMOS 晶体管 384 截止。当 PMOS 晶体管 384 截止时,因 PIPECNT1 变为高电平,在下一个时钟脉冲通过 NMOS 晶体管 390 将 PIPECNT0 快速拉到地电位。PIPECNT0 选定高电平使逻辑 1 被送入与非门 362 的输入端。或非门 334 给与非门 362 的另一输入端提供 1。所以与非门 362 将输出逻辑 0。经反相器串 364 的延时(延时选取大于 Clkt2 上的脉宽)之后,在下一个时钟脉冲的传播中将把节点 16 和 NMOS 晶体管 366 的栅极拉到高电平。同样在下一个时钟脉冲前,由于 PIPECNT0 被选在高电平,节点 N015,NMOS 晶体管 372 的栅极被拖到低电平,将或非门 336 的输出拖到低电平。因为晶体管 372 截止,下一个时钟脉冲将不影响节点 N21。然而,因为晶体管 366 导通,将把节点 N22 拖到低电平。PMOS 晶体管 402 快速使 PIPECNT1 选定高电平。PIPECNT1 的选择高电平导致 NMOS 晶体管 390 使 PIPECNT0 选择低电平。

PIPECNT 信号的快速正转换部分是通过使用具有大的(即  $1000\mu$  量级)的信道宽度的 PMOS 晶体管 384、402 和 420,快速提升与导线相关的比较高的电容并逻辑耦合到每个 PIPECNT 信号而实现的。为进一步降低延时,在 PMOS 晶体管开始将各节点拉到高电平之前,与它们相反的 NMOS 晶体管 390、408 和 426 早已截止。即,比如在 PIPECNT1 (NMOS390 的栅极)为低电平时,PMOS 晶体管 384 将 PIPECNT0 拉到高电平。NMOS 晶体管 390、408 和 426 也提供比较大的信道宽度,以实现快速负转换。另外,因为 PMOS 晶体管对宽度有限的脉冲是导通的,当 PMOS 晶体管早已截止时,这些 NMOS 晶体管导通。再有,为了实现快速运行,Clkt2 上的电容由于只要求 Clkt2 驱动单个晶体管,NMOS 晶体管 344 而被降至最小。如果计数器是用三个常用与非门构成的,每个各产生一节点 N21、N22、N23,则三个分离的 NMOS 和三个 PMOS 晶体管应与节点 Clkt2 相连接。这显然会增加 Clkt2 的电容。

时钟与新的 PIPECNT 的选定之间的很短的延时得到由时钟脉冲至输出的很低的存取时间。如图 3 和 4 的实例定时图所示,由流水计数电路 330 所产生的信号有快的转换,即在 Clkt2 转换为高电平之后只经两个反相延时转换高电平,同样也是在外接收时钟之后只经两个反相延时转换高电平。在 1 时钟周期选择 PIPECNT0—PIPECNT2 中每一个信号为高电平,在 2 时钟周期选择低电平,而其余两个轮换被选择。用这些信号与图 2 的导流电路 200 相结合,来选择特

定的一对寄存器,即支路 0 的存储节点 N01 和 N04 中所存储的数据,表现为来自存储器件的输出数据。流水计数电路是自启动的。在任一周期,如果 PIPECNT0 和 PIPECNT1 均未被选中(不管 PIPECNT2 是否被选中),或非门 336、与非门 368 延时串 370 和晶体管 372 将导致下一周期 PIPECNT0 的选择。在以后各周期接着是 PIPECNT1、PIPECNT2、PIPECNT0、PIPECNT1 等等。

现在参照图 8 描述,使用 PIPECNT 信号与流水计数延时电路 440 的结合。如上面结合图 1 和图 2 所描述的,为把数据从本发明的 I/O 线导流到存储节点 N01—N06 中置位高电平的一个。具体来讲,在 I/O 或 I/O# 线上必须出现低脉冲,而单一的 PIPECNTDEL 信号必须择用低电平。如果 PIPECNTDEL0 择用低电平,数据将被导流到支路 0 的节点 N01 或 N04。如果 PIPECNTDEL 或 PIPECNTDEL2 择用低电平,数据将分别被导流到支路 1 的节点 N02 或 N05 或者导流到支路 2 的节点 N03 或 N06。采用流水计数延时电路 440,确保在适当时间出现 PIPECNTDEL 信号并导流到适当的支路上。在高频时,这是个具体的难题。在一具体实施例中,每个 PIPECNTDEL 信号在读取周期开始后大约 8.5ns 递增,并根据选取的等待间隔跟随(变换)一个特定的 PIPECNT 输出。

流水计数延时电路 440 根据包括指示预期的等待间隔的信息的几个输入产生 PIPECNTDEL 信号,如果采用等待间隔为 1,整个电路都是不必要的,因而被禁止。通过使信号

CL1 维持高电平,而使该电路禁止。这是因为反相器 442 输出一个低电平信号,因而使 PMOS 晶体管 526、528 和 530 导通。这也迫使与非门 444 输出一个高信号,并使非门 446 输出低信号,使 NMOS 晶体管 514、518、522 截止。非门 448 的高输出使 PMOS 晶体管 516、520 和 524 截止。当晶体管 514 和 516 截止,而 526 导通时,非门 534 的输出为高电平,而非门 538 和 542 的输出照旧。这迫使信号 PIPECNTDEL0—2 固定在其非择用的高电平,以使导流电路 200 永不将数据从 I/O 线导流到节点 N01—N06。

相反,当采用等待间隔为 2 或 3 时,将 0 输入到线 CL1。可采用模式寄存器等可编程地规定等待间隔为 2 或 3。例如,在一具体实施例中,在 merg4 线上的逻辑 0 选定等待间隔为 2,而逻辑 1 选定等待间隔为 3。如果选定 2,将启动由晶体管 478/480、486/488 和 494/496 形成的多路开关的上半部。如果选定等待间隔为 3,则启动由晶体管 482/484、490/492 和 498/500 形成的多路开关的下半部。可以采用其它的选择电路用于具有更多支路的执行程序,来增加具有更多的合理的等待间隔的电路。存储设计可按下面完成,允许用户通过改写模式寄存器等内部的一个值来改变已装入部分的等待间隔。

从图 7 的 PIPECNT 电路 330 接收信号 PIPECNT0 和 PIPECNT1。当 PIPECNT0 或 PIPECNT1 选择高电平时,由或非门 564 产生一个替代 PIPECNT2 信号并被选定高电平。在 Clkt2 脉冲之后很快将信号 PIPECNT0、PIPECNT1 或

PIPECNT2 的状态改为高电平,以实现从时钟脉冲至输出的快速存取时间。Clkt2 脉冲被图 7 的反相器串 350 延时,再被图 8 的反相器串 562 延时,大约在 Clkt2 后的 3.3ns 提供一正脉冲。延时通过电路的存储部分沿着数据脉冲的非同步传播延时路径约等于延时的三分之一左右。在 PIPECNT 信号结束递增之后发生此脉冲,并在 5ns 之后,再次递增之前终止。此正脉冲使每个 PIPECNT 导致新的状态,通过由晶体管对 452/452、456/458 和 460/462 所形成的多路开关传播。在 Clkt2 信号已输入到图 7 的流水计数电路之后大约 3.4ns 在多路开关的计数数据输出递增。大约 0.1ns 后非门 468、472、和 476 将此新的数据传播给节点 N25、N26 和 N27。即,由于延时串 352 和 562 以及通过多路开关 452/454 及反相器 468 等,使节点 N25—N27 落后 PIPECNT0、1、2 的反相转换 3.5ns,通过多路开关的信号被由反相器对 466/468、470/472 和 474/476 形成的锁存器保持。在节点 N25—N27 的数据会非同步地传播到节点 N28—N30,但要根据所选定的等待间隔(确定)哪一种输入 N25—N27 产生给定的输出 N28—N30。比如,选定等待间隔为 2,来自节点 N25 的信号将传至,节点 N28,节点 N26 上的信号将传至节点 N29,而 N27 上的信号将传至节点 N30。即,在适当延时后,因 PIPECNT0 选择高电平,信号 PIPECNTDEL1 将择用低电平。PIPECNTDEL2 将由 PIPECNT1 的选择引起,而 PIPECNTDEL0 将服从 PIPECNT2。当使用此种信号方案与图 2 的导流电路 200 结合时,其结果如图所示,根据在读取周

期开始后接收的第一个时钟脉冲,将数据送到输出缓冲级 150。

当选定等待间隔为 3 时,在节点 N25—N27 的信号将通过多路开关的下半部。即,来自 N25 的数据传至节点 N29,来自 N26 的数据将传至节点 N30,而来自节点 N27 的数据将传至节点 N28。在此情况下,提前 8.5ns 将由 PIPECNT2 的选定维持信号 PIPECNTDEL1 为低电平。PIPECNTDEL2 服从 PIPECNT0,而 PIPECNTDEL0 将服从 PIPECNT1。这就确保在读取周期开始后两个时钟脉冲,即以等待间隔为 3 由电路 200 将数据放到输出缓冲级,如图 4 所示。参照图 3 和图 4 的定时图,可以看到选取不同等待间隔的效果。在图 4 中,选取等待间隔为 3,提前 8.5ns 由信号 PIPECNT1 的选择高电平,将数据 PIPECNTDEL0 维持在低电平。时钟脉冲 0 启动此读取周期,时钟脉冲 1 从支路 2(如果存在)选择数据,而时钟脉冲 2 从读取数据导流至支路 0 选择数据。在图 3 中,选取等待间隔为 2,提前 8.5ns 由信号 PIPECNT2 的选择高电平,使信号 PIPECNTDEL0 维持低电平。在此情况,时钟脉冲 0 启动此读取周期,并将数据导流到支路 0。时钟脉冲 1 为输出在支路 0 中选择此数据,实现等待间隔为 2。这种信号路径的安排允许在不同等待间隔之间的很有效而简单的选择。当然,可以使用其它逻辑装置选取服从(哪个) PIPECNT 的那个 PIPECNTDEL。比如,可以使用组合逻辑,如与非门或者或非门按类似方式来选择信号。

一旦适当信号达到节点 N28、N29 或 N30,使用另一组

多路开关(由晶体管对 514/516、518/520 和 522/524 形成)通过电路的其余部分而递进,定时受定时信号 COUNTDEL 控制。将此定时信号设计成,在由存储器产生读取请求后,择用下一个 PIPECNTDEL 信号择用前一个已择用的 PIPECNTDEL 信号适当时间。在优选实施例中,这些信号转换稍微早于数据到达 I/O 线。当在高频运行时,精确定时是绝对需要的。下面的电路提供此类系统所需要的精确定时。图 9 中表示用于产生 COUNTDEL 信号的定时电路的一个具体实施例 600。

用于将 I/O 数据导流到电路 200(图 2)各个支路的 PIPECNTDEL 要求精确定时。此定时必须与在 I/O 线上负数据脉冲一起的出现相一致。当选定(DRAM 的)列解码器输出时,存取晶体管把所选定的列的 bit 线和 bit<sup>#</sup>线耦合已与相同电压平衡的局部 I/O 和 I/O<sup>#</sup>线。bit 线或 bit<sup>#</sup>线之一是高电平,而另一个是低电平。所得到的在不同电压下对节点不均匀导电引起差分电压,在真实的和等效的局部 I/O 线之间缓慢生成,在生成充分模拟信号后,用一动态差分读出放大器 622 选通由此种差分电压所代表的的数据。读出放大器的选通脉冲是个窄脉冲,根据从局部 I/O 或 I/O<sup>#</sup>接收的差分电压的极性,使读出放大器在真实的或等效的(全程)I/O 线上输出窄的负脉冲。在选通使读出放大器有充足的信号以可靠运行之前等待足够长(的时间)是至关重要的。但不等待长于会增加存储器的地址访问时间所需(的时间)。为达到精确读出放大器选通脉冲定时,按下述方式产生定时,接近实际,

在维持选定的列后固定的延时,生成引起局部 I/O 差分信号  
的信号。在一具体实施例中,采用电路 600 来控制这种固定  
的延时。

在一具体实施例中,定时电路 600 接受八个预解码器的  
输出,它们是在由四个或非门 602—608、两个与非门 610、  
612 及紧跟着的单一或非门 614 组成的逻辑或非电路中的  
一起进行“或”运算的 GYA0—GYA7。本领域的技术人员将  
了解,把许多列解码器的输出“或”起来以产生读出放大器的  
选通脉冲定时信号是不实际的。然而,将八个 1 置位一起  
“或”成八中取一预解码输出,其中的脉冲高电平之一稍早于  
最后列解码器输出脉冲高电平是可行的。本领域的技术人员  
还应认识到,如果所选定的线脉冲是低电平而不是高电平,  
也可用逻辑与非电路来检测预解码线中的信号。

当任一公用 Y—地址产生高脉冲时,或非门 602—608  
之一的输出产生低脉冲,导致与非门 610 或 612 产生高脉  
冲,从而迫使或非门 614 产生低脉冲输出。在此之前,或非门  
614 的输出已为高电平,反相器的输出已为低电平。于是当  
或非门 614 的输出转换为低电平时,对一短周期(等于反相  
器链 616 的延时),或非门 618 输出高脉冲。此脉冲比如可以  
是个 1ns 宽脉冲。选取这种窄脉冲,以允许 I/O 线上的窄数  
据脉冲。I/O 线上的窄数据脉冲改善了定时裕度,如将要描  
述的。在择用预解码器输入之一后,几乎与择用最后列解码  
一致,此脉冲的读取沿可发生四个逻辑延时。

用缓冲器和选择逻辑 620 来缓冲和逻辑选择



COUNTDEL 输出上的正脉冲;以便仅选通选定的存储器阵列的选定读出放大器。因为读出放大器选通脉冲是被与择用列的同一信号(GYA0—GYA7之一)择用的。它们相对于列选择的定时是很好的。在择用列之后 2.5ns 产生读出放大器选通脉冲信号,以允许生成差分信号的时间。在读出放大器及其输出缓冲器的延时后,在 COUNTDEL 上的正脉冲之后的 4ns 在真实的或等效的(全程)I/O 线上产生 2ns 宽负数据脉冲。

如前所述,PIPECNTDEL 信号必须很精确定时地递进。为达到所要求的定时精度,采用两种独立的概念。首先,在 I/O 上出现数据作为脉宽尽可能短的脉冲。这种短脉冲使时间窗口最大,在该窗口中可增加 PIPECNTDEL 而不会招来误差。对 2ns 脉冲和 5ns 的周期时间,该窗口是 3ns 宽。如果读出放大器代之以输出一个逻辑电平(与一个脉冲对照),在准确与数据递增准确相同的时间会使 PIPECNTDEL 有个递增,不为误差留下余地。其次,也使用使读出放大器选通脉冲定时(在 I/O 或 I/O# 上产生负脉冲)置位的同一 COUNTDEL 信号来使 PIPECNTDEL 信号递进的定时置位。当然,也能使用除 COUNTDEL 以外的信号。对数据的正当导流极为重要的是使用同一信号来产生 I/O 脉冲和使 PIPECNTDEL 递进。

再参照图 8,可以看到,COUNTDEL 脉冲被用于使与非门 444 运行。如果未选定等待间隔为 1,并假定 COUNTDEL 信号产生高脉冲,与非门 444 将输出一负脉冲,然后被反相器 446 反相。于是,在反相器 446 的输出端产生脉宽可能为

1ns 的正脉冲。该正脉冲导致由晶体管对 514/516、518/520 和 522/524 形成的多路开关让来自节点 N28—N30 的数据传播通过,变成信号 PIPECNTDEL1、2 或 0。即,如果 PIPECNTDEL0 被维持低电平,从与非门 512 输出逻辑 0,通过由晶体管 522 和 524 形成的多路开关传播(在特定时间)由非门锁存器 540/542 锁存,由非门 542—560 缓冲,作为 PIPECNTDEL0 输出。从 COUNTDEL 至 PIPECNTDEL0(或 PIPECNTDEL1 或 2)的总延时约为 2.5ns。这是通过与非门 444、非门 446、NMOS 晶体管 522 以及非门 542、556、558 和 560 的延时。具体选取 2.5ns 延时要短于从 COUNTDEL 至窄 I/O 数据脉冲的读取沿的 4.0ns 延时。于是稍稍在 I/O 线产生低脉冲前,使 PIPECNTDEL 信号增加。PIPECNTDEL 信号有比较长的持续期,而维持对全时钟周期时间有效。这是以增加 I/O 脉冲可达到的窗口。

可把在图 8 电路 440 的一具体实施例的各节点的定时归结如下。PIPECNT0、1、2 在时钟脉冲的每个上升沿之后快速递进。在时钟脉冲的上升沿之后大约 3.3ns 启动多路开关 452/454 等。在多路开关 452/454 等启动之后大约 1.2ns 或在时钟脉冲的上升沿之后大约 4.5ns,与非门 504/512 的输出增加。在一个独立的信道中,信号 COUNTDEL 产生正脉冲,在择用时钟之后近似 6ns 上升,并在 1ns 后下降。非门 446 在时钟脉冲后 6.5ns 开始输出一个正 1ns 宽脉冲。在此时,启动多路开关 514/516 等,传送来自与非门 504、508、512 数据,提前 2.0ns(时钟脉冲后 4.5ns)变为有效。最后在

多路开关 514/516 等启动后的 2ns,在时钟脉冲后 8.5ns 使 PIPEDNTDEL 输出递增。最重要的是,它们在 2ns 宽 I/O 脉冲达到前 1.5ns 递增。对于 5ns 的周期时间,在 5ns 后,或在 2ns 宽数据脉冲的末尾之后大约 1.5ns 使 PIPECNTDEL 信号再次递增。因而,在 5ns 周期时间,每个 PIPECNTDEL 对 I/O 数据脉冲的两侧的全部 1.5ns 是有效、择用或放弃的。窄的 2ns 宽的脉冲使种 1.5ns 定时裕度最大。由于在有功的 I/O 线的远端上升、下降时间相当慢,此脉冲是 2ns,而不是 1ns。在非门 446 的输出端在 1ns 宽脉冲的两边的全部 2ns,非门 504—512 的输出是有效的。所以在与非门 504、508、512 的输出数据的达到应早或迟 2ns,而在 PIPECNTDEL 定时内无变化。这些裕度允许 PIPECNTDEL 信号定时的协调,并使此种流水线的方法实用。

在图 8 的电路 440 的一具体实施例中,可以用读信号使与非门 504、508 和 512 的输出禁止。即,如果读取周期无效,送到信号线 READ 上的低信号将确保所有的与非门 504、508 和 512 输出逻辑 1。将设有 PIPECNTDEL 信号择用低电平,因而,切换这些信号将不消耗功率。

现在参照图 10,表示根据本发明的一个复位电路的具体实施例。采用流水线复位电路 630 来产生输入到本发明的导流电路 200 的 RESET-2 信号。RESET0 信号(正脉冲)将使(图 2)电路 200 的节点 N01 和 N04 复位为低电平,而使节点 N013 复位为高电平。在周期内早期出现支路 0 的复位,跟随来自支路 0 的循环输出数据。即,在不再为输出选择支

路 0 之后在用于周期选择支路 1 的早期,立刻出现,但在从 I/O 或 I/O# 将新数据导流型至支路 0 之前,终止该 RESET 脉冲。由在周期的早期输出正脉冲的时钟信号 Clkt2 驱动电路 630。由反相器链 644 增加延时。应选择反相器链的数目,以确保在流水计数已递进到下一个计数之后产生复位信号。比如,图 2 的电路 200 的存储节点 N01 和 N04 需要被复位到低电平,在某些点上,在选择 PIPECNT0 后,但必须在新数据被导流将节点 N01 或 N04 置位为高电平之前使复位终止。在一具体实施例中,复位电路 630 是这样协调的,大概在时钟脉冲接收之后的 3ns 在反相器串 644 的输出端出现正脉冲。在此时,信号 PIPECNT0、1 或 2 之一刚好已被选定为高电平。于是与非门 632—636 之一的输出端将产生低脉冲。比如,如果刚好 PIPECNT1 被选定为高电平,与非门 632 将输出低脉冲,而 RESET0 将输出高脉冲,这种脉冲在新数据被导流到支路 0 内之前终止。其余的 RESET 信号(1 和 2)在本周期内不产生复位脉冲。按类似方式,PIPECNT2 信号的选择将导致 RESET1 信号的产生,而 PIPECNT0 的选择将产生 RESET2 信号。可以调 RESET 信号产生的相对定时。然而,RESET 信号必须出现在其支路不再为输出选择之后并在新到达其支路前终止。

当已选定等待间隔为 1 时,非门 631 提供低输出,而每个 RESET0—2 将有高输出。在此情况下,图 2 的禁止电路 200 的所有节点 N01、N02、N03、N04、N05 和 N06 仍维持复位低电平。无复位信号改变电压,因而节省电源。

大多数存储电路有个要求,有时,比如当与此芯片并行地选择其后相同的存储芯片时,要求输出端开路。确定一给定的时钟周期是用于输出数据还是用来提供开路条件的逻辑对本领域的技术人员是公知的。对一本发明的目的,需要此信息在它生效的周期开始之前变为有效。在周期末尾维持(高电平)时,称为 OUTEN(输出启动)的信号导致在下一个时钟脉冲输出下一个有效数据,如上所述。但是,如果禁止,根据接收下一个时钟脉冲,OUTEN 使输出快速开路。为得到开路输出,图 6 的电路 300 中的 OUTOFF 择用高电平。此高电平被反相器 302 反相,以使为 PMOS 晶体管 304 产生低输入。这就把节点 PULLUP 拉到高电平,使输出 PMOS 晶体管 314 截止。OUTOFF 上的高电平导通了 NMOS 晶体管 316,以便把节点 PULLDN 拉到低电平,使输出 NMOS 晶体管 318 截止。当晶体管 314 和 318 截止时,输出端是开放电路。

为了节省电源,没有晶体管把 PULLUP 拉到低电平,或把 PULLDN 拉到高电平。暂时返回到图 2 的电路 200,当 PIPECNT 递进(比如说从 PIPECNT0 到 PIPECNT1)时,IPPECNT1 快速选择高电平,使支路 0 的 PMOS 晶体管 230A 和 244A 截止,而 PIPECNT0 快速放弃低电平,也使支路 0 的 NMOS 晶体管 234A、226A 和 240A 截止。即,防止电路 200 先前所选定的支路 0 连续把 PULLUP 拉到低电平,或把 PULLDN 拉到高电平。但还要求,在 IPECNT1 选择期间,没有数据来自支路 1,在这种输出禁止周期,试将 PULLUP 拉低,或将 PULLDN 拉高。在这种输出禁止周期,即使选择

PIPECNT1 为高电平,通过保持 PCNT1 在低电平来达到此点。必须将 PIPECNT1 选定为高电平,以便如前所述,使先前所选择的支路的 PMOS 晶体管 230A 和 244B 截止。在开路输出的周期内,尽管 PIPECNT0—2 的一个信号被选择高电平,所有 PCNT0—2 均为低电平。PCNT0—2 信号被图 11 的电路 650 中的 PIPECNT0—2 信号驱动。如后所述,如果在一个周期末尾 OUTEN 是高电平,信号 PCNT0—2 将分别跟随信号 PIPECNT0—2,在下一时钟为高电平。这为大于 1 的等待间隔提供先前描述的运行。然而,如在一个周期的末尾 OUTEN 为低电平,下个周期当其对应的信号 PIPECNT0—2 被选择为高电平时,信号 PCNT0—2 将保持低电平。

在将 PIPECNT0 选择为高电平的周期的末尾,OUTEN 为高或低电平,以便为下个周期分别启动或禁止输出。如果 OUTEN 为低电平,而 PIPECNT0 为高电平,晶体管 668 使弱反相器 674 过载,将 N42 近到低电平。在此时 NMOS 晶体管 672 截止,而反相器把节点 N43 拉到高电平。反相器 674、676 锁定,并在 PIPECNT0 转换为低电平后保持此条件。如果,代之以,OUTEN 为高电平,而 PIPECNT0 也为高电平时,NMOS 晶体管 670 和 672 均导通,使反相器 676 过载,并将节点 N43 拉到地电位。反相器 674 将节点 N42 拉到高电平,在 PIPECNT0 转换低电平后,锁存并维持此条件。因而,当在下一时钟脉冲 PIPECNT0 变为低电平时,由反相器 674 和 676 组成的锁存器可以不再改变状态,但刚好在 PIPECNT0 切换到低电平之前代之以保持 OUTEN 值的数据特征。如果

当 PIPECNT0 转换为低电平时, OUTEN 为低电平(当择用 PIPECNT1 时, 使输出禁止), 节点 N42 为低电平, 而 N43 为高电平。为 N42 为低电平, 使 NMOS 晶体管 678 截止, 为 N43 为高电平, 也使 PMOS 晶体管 680 截止。所以在 PIPECNT1 和 PCNT1 之间没有联系。同时, 当 N43 为高电平, 使 NMOS 晶体管 688 导通, 保持 PCNT 为低电平。因而在在前个周期的末尾 OUTEN 为低电平时, 在下一个周期 PCNT1 不跟随 PIPECNT1 的高电平, 因而在(图 2)电路 200 的输出端设有出现来自支路 1 的新数据。当 PIPECNT1 变高电平时, 其它电路(未图示)驱动(图 6)电路 300 的 OUTOFF 到高电平。这就把 PULLUP 拉到高电平, 把 PULLDN 拉到低电平, 使输出端开路。

另一方面, 如果在前个周期(当 PIPECNT0 为高电平)的末尾 OUTEN 为高电平, 那么节点 N42 为高电平, 而节点 N43 为低电平(图 11)。在此情况下, 晶体管 678 和 680 导通, 晶体管 688 截止。所以, PCNT1 跟随 PIPECNT1 的高电平, 而在下一个周期的末尾跟随 PIPECNT1 的低电平(PIPECNT0 仍为低电平)。当 PCNT1 选择高电平, 来自(图 2)电路 200 的置位存储节点 N02 或 N05 的数据驱动节点 PULLUP 和 PULLDN 同高或同低, 以便输出数据。按类似方式, 当选择 PCNT0 和 2 时, 它们各自跟随或不跟随 PIPECNT0 和 2。

因而, 电路 650 提供一个有效而简明的电路, 用于当选定其 PIPECNT 信号时启动电路 200 的一个支路, 以提供输出数据, 或当选定其 PIPECNT 信号时, 禁止该支路。

虽然上面是本发明的一具体实施例的完整的说明,但可采用各式各样的改型。比如,系统可以设置另一电路分支,允许从具有等待间隔为1—4的存储系统产生输出。还有,可以重复本文所述的电路,以支持任何次数的等待间隔。该流水线可以与若干同步存储器产品一起使用。该电路也能适用于正和/或负沿触发。虽然在几个实例中提及了具体的信道宽度,这些参数仅仅意味着与本发明的一个具体实施例有关。本领域的技术人员,通过阅读此说明,将能改变信道宽度和晶体管类型,以适应特别需要。

所以,本发明的公开意在解释,而不是对本发明的在所附权利要求中提及的范畴的限制。



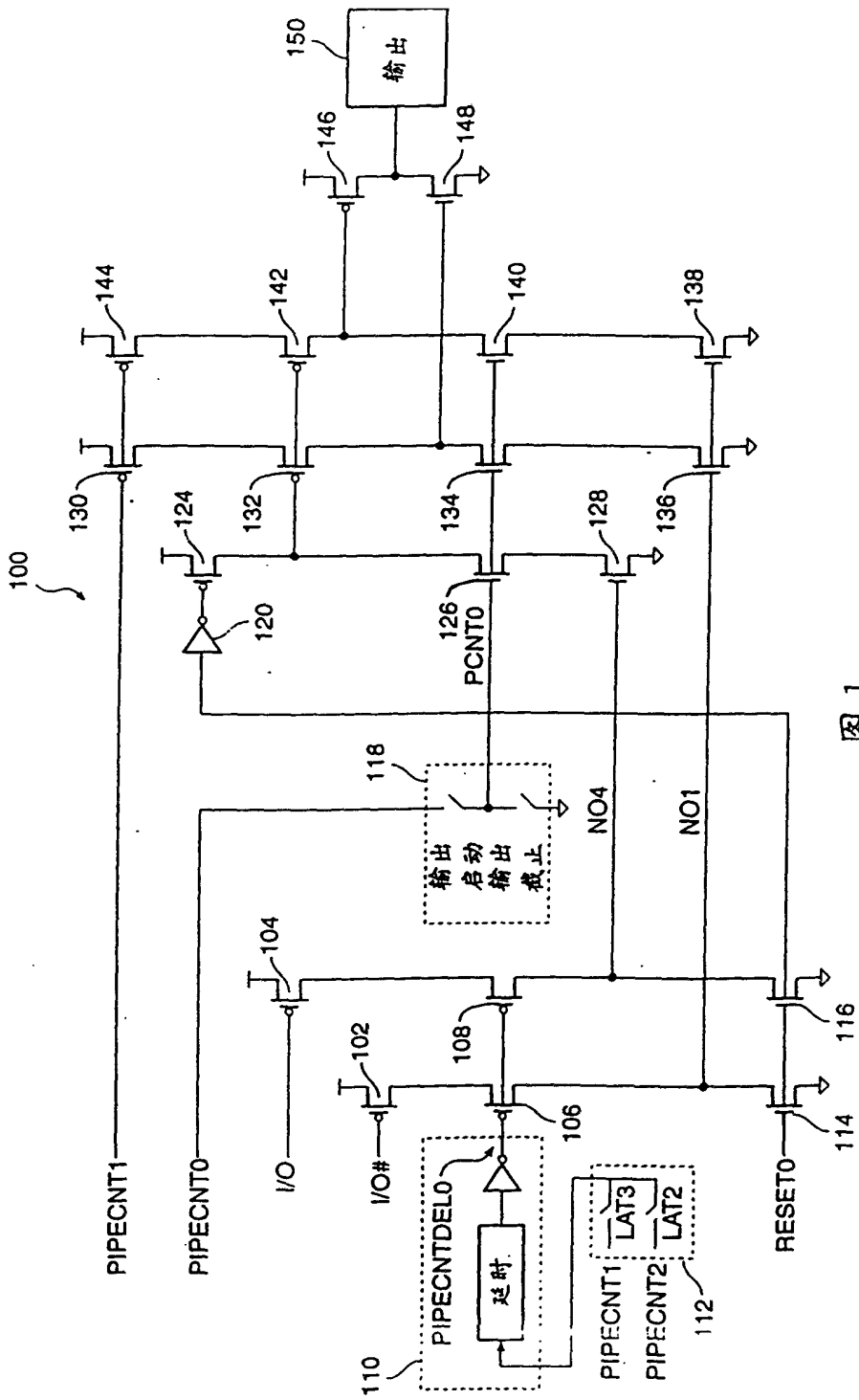


图 1

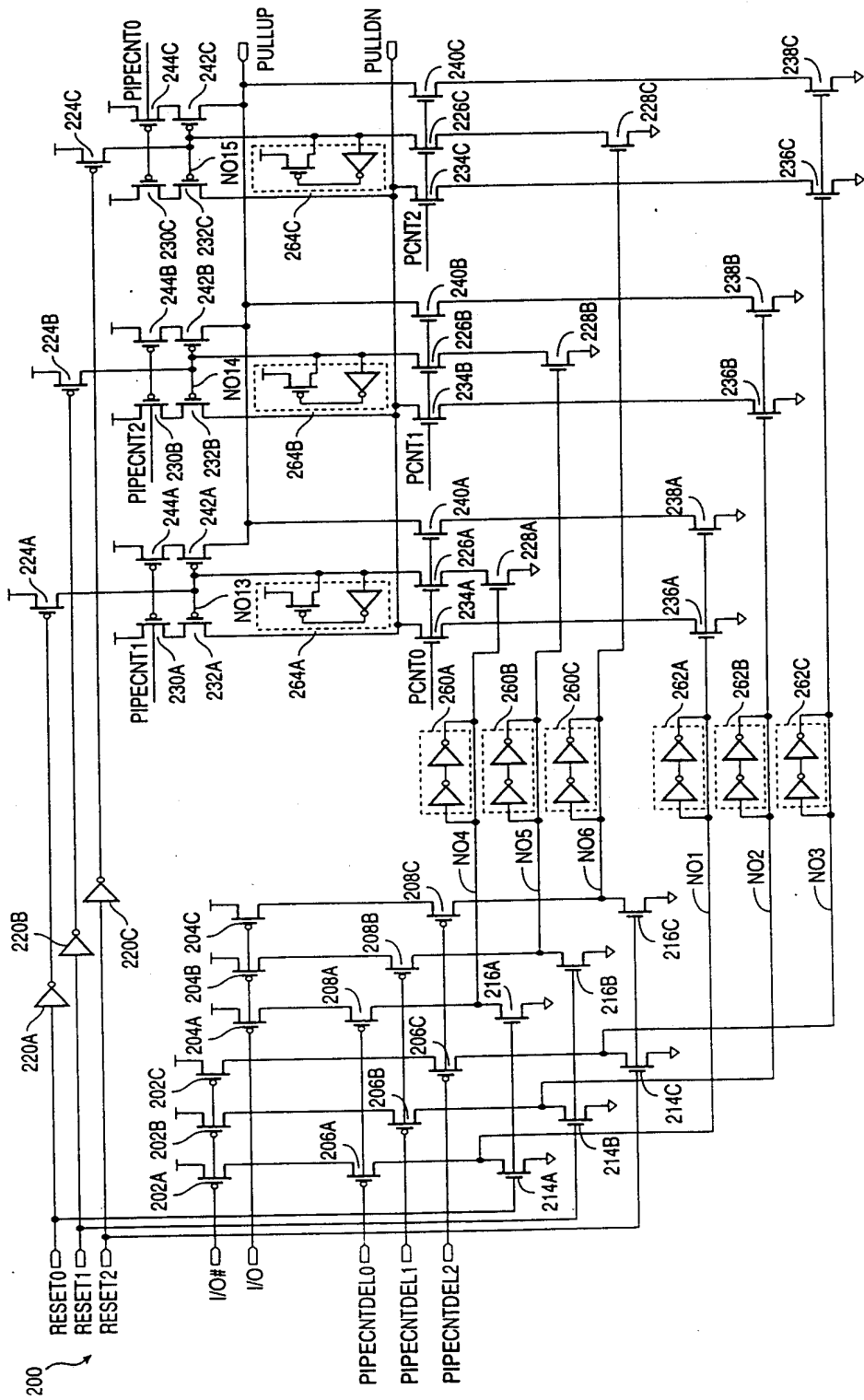


图 2.

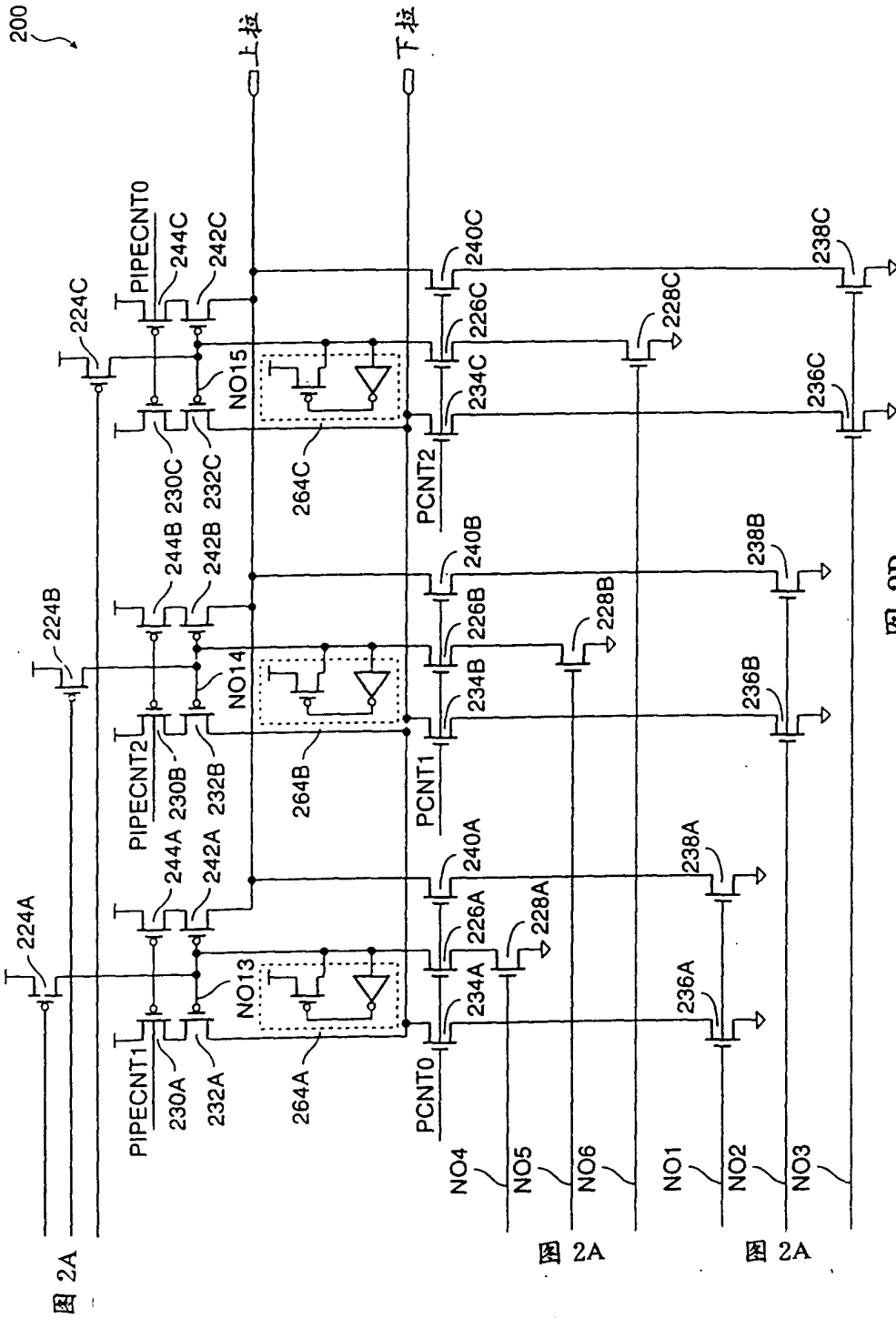


图 2B

图 2A

图 2A

图 2A

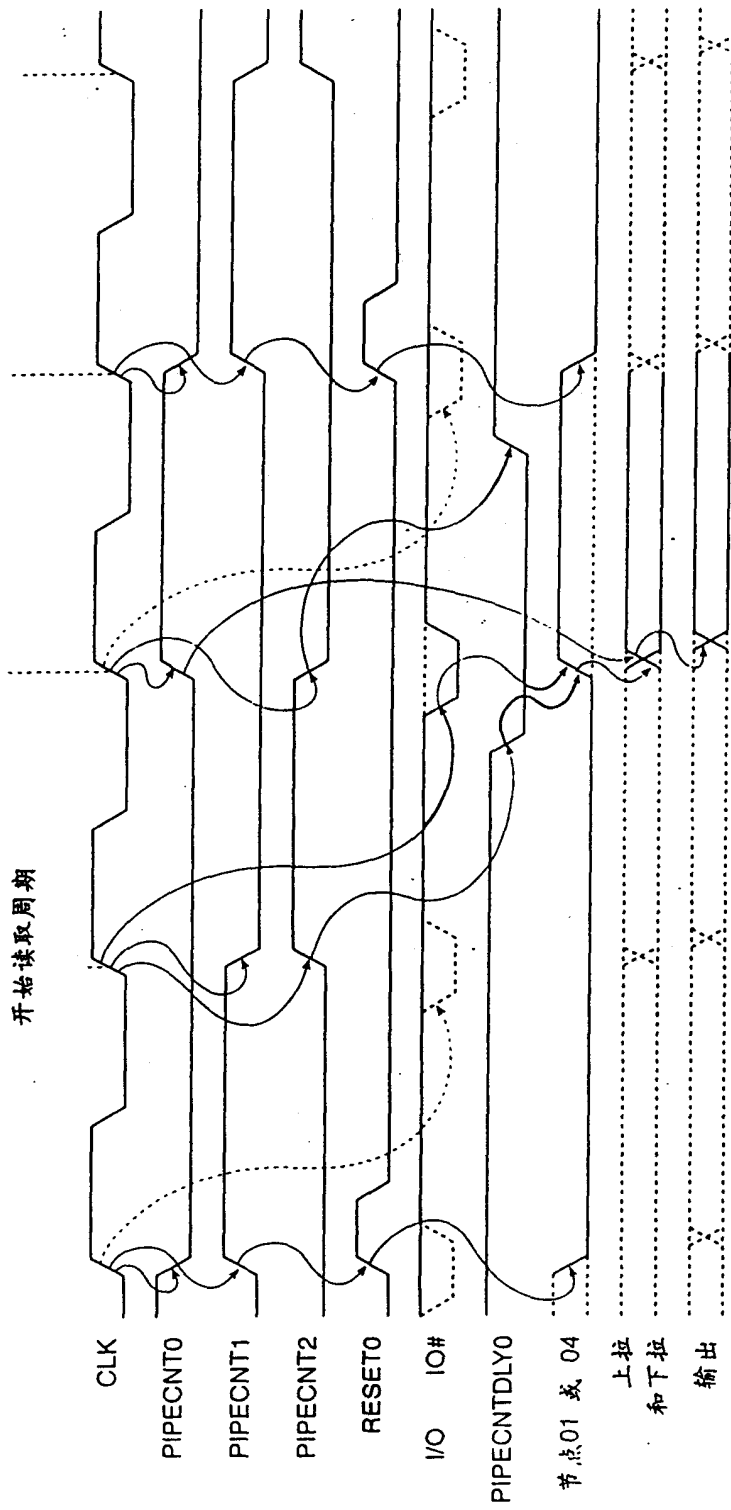


图 3

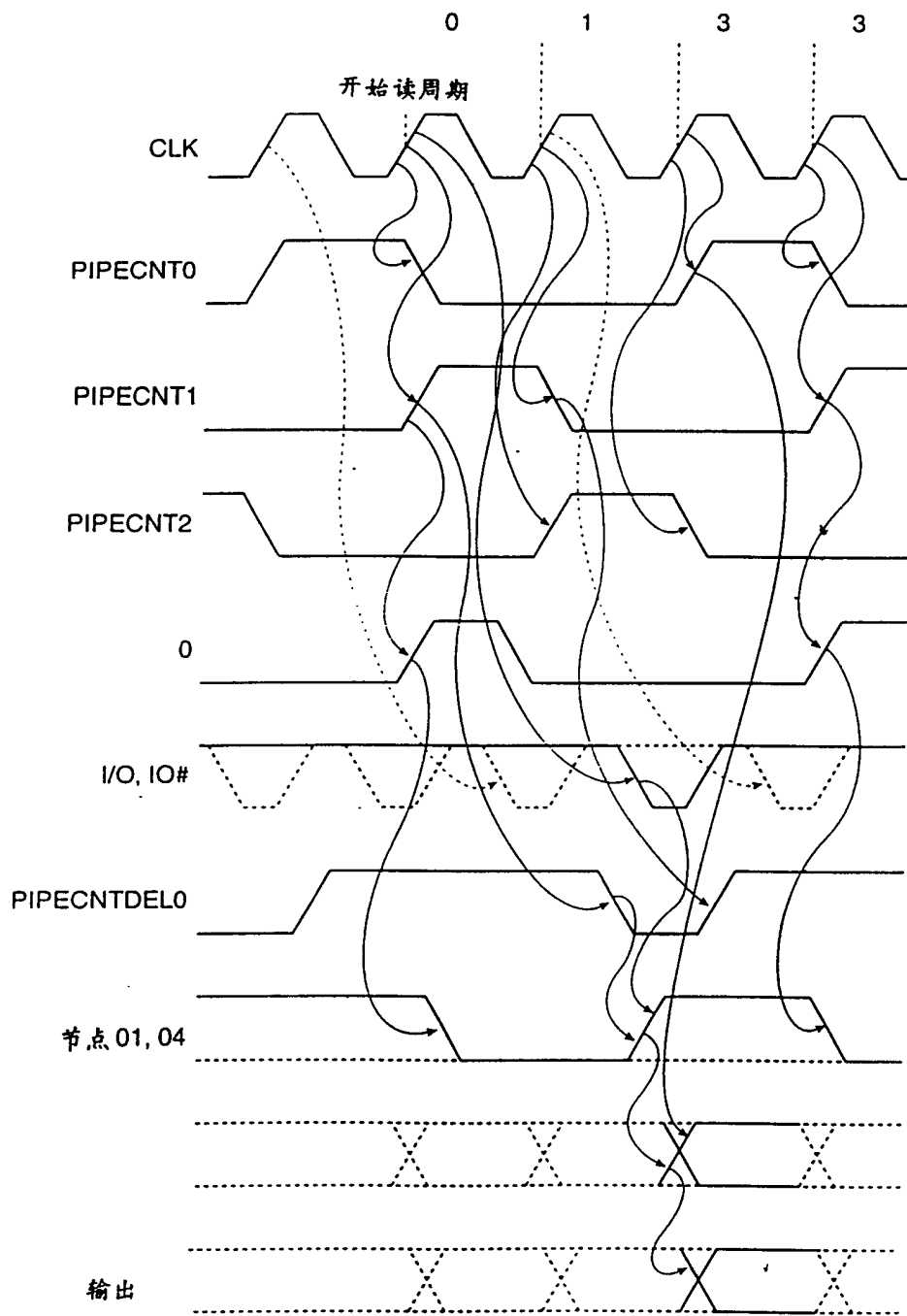


图 4

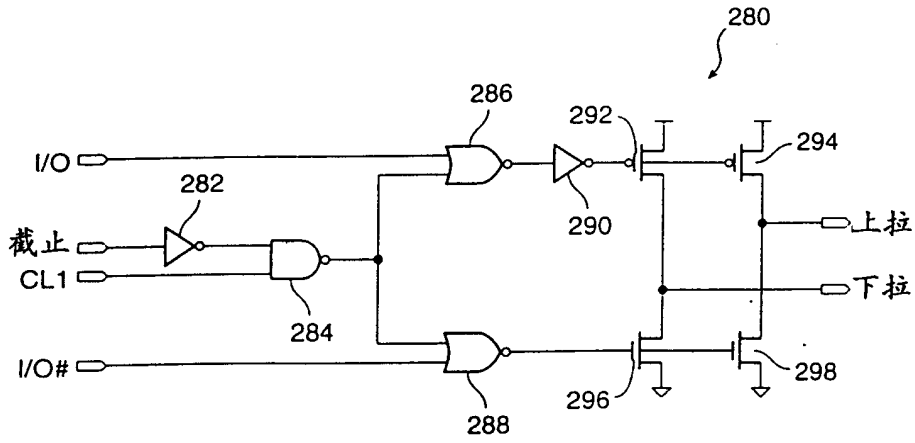


图 5

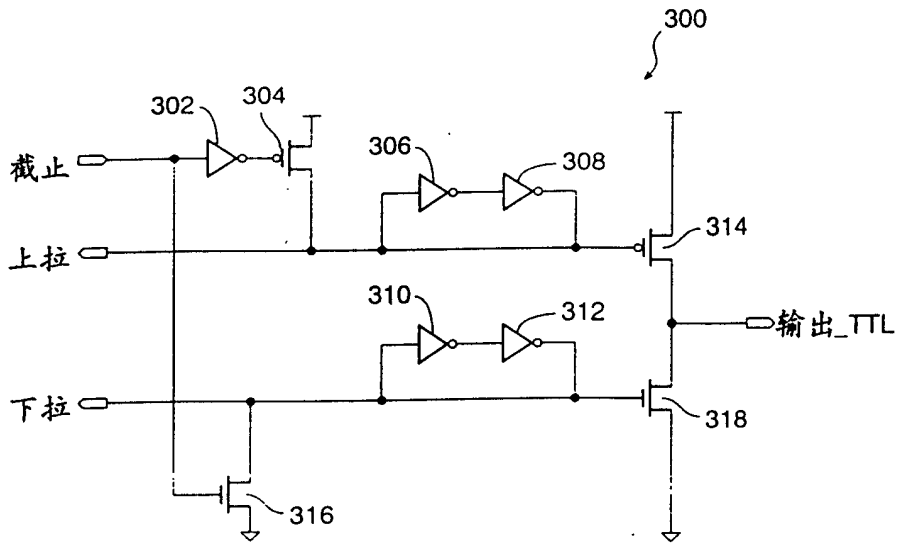
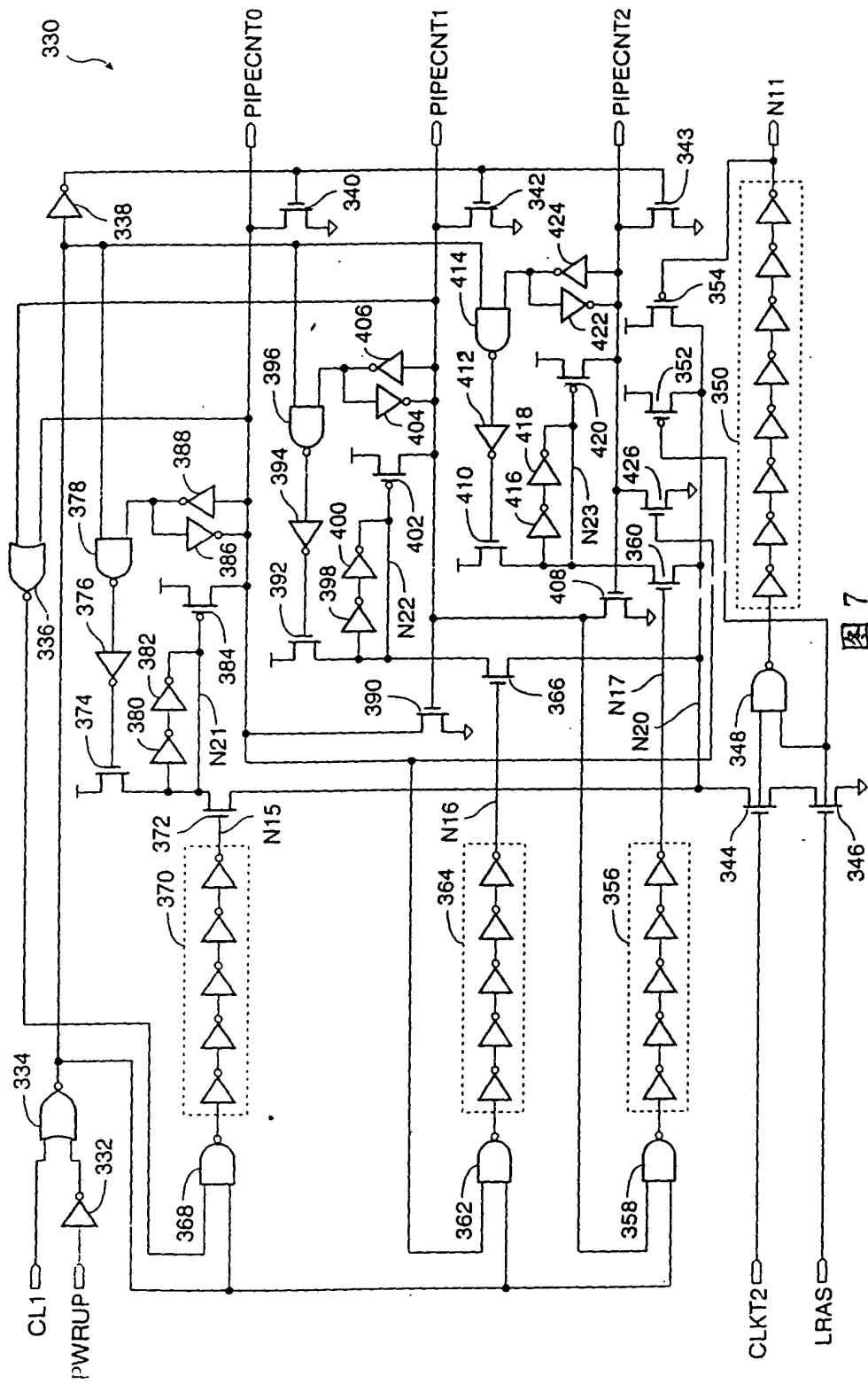


图 6



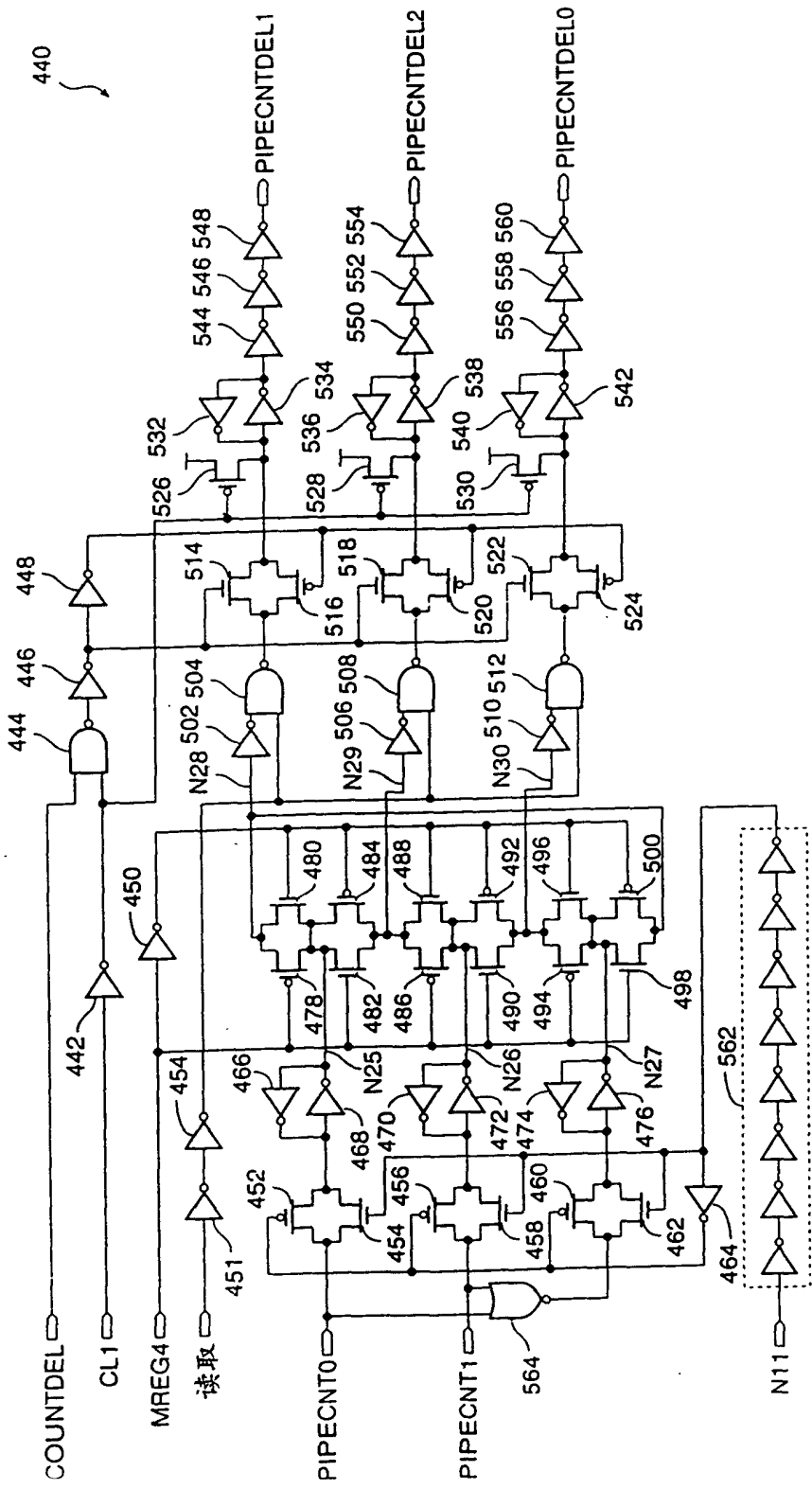


图 8



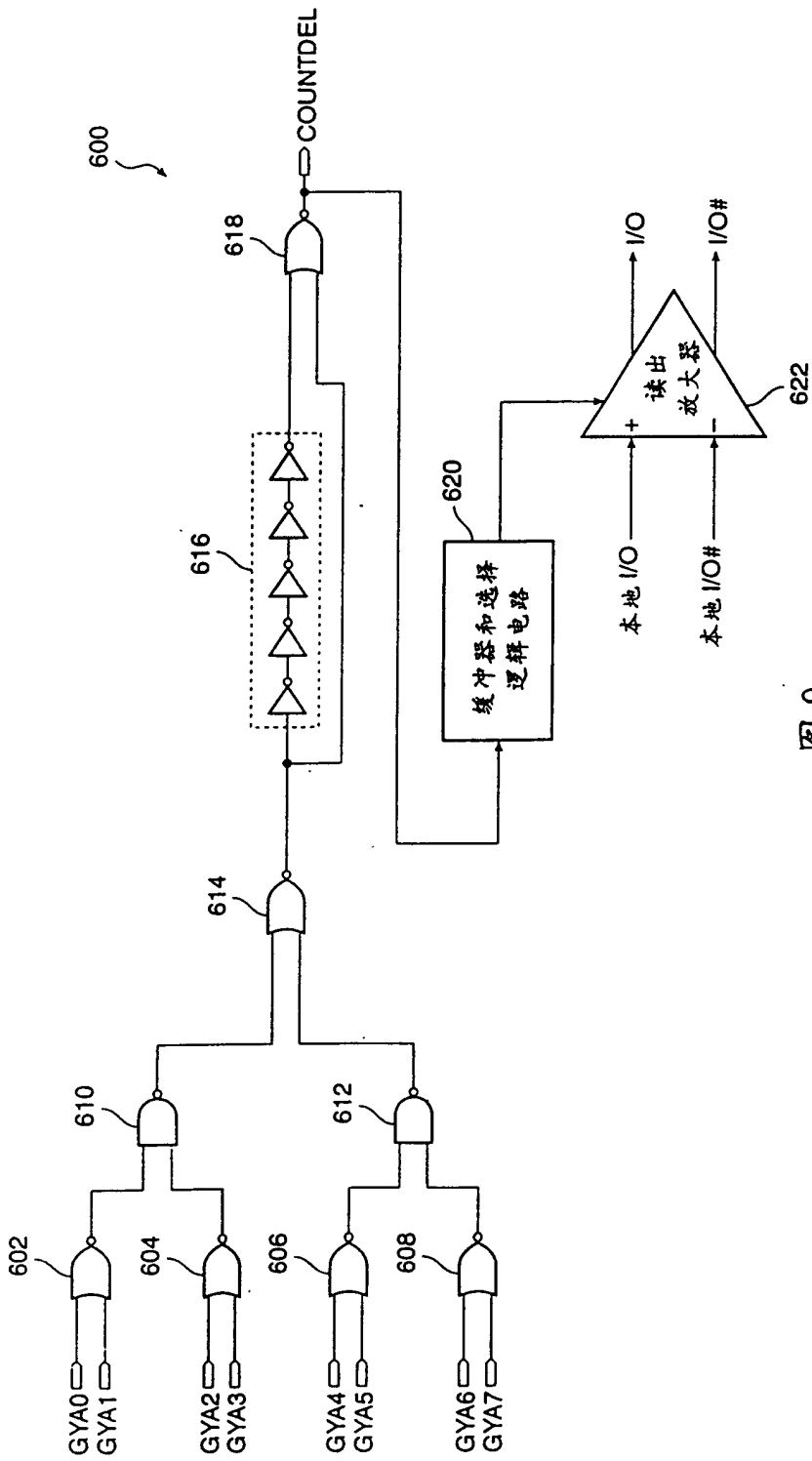


图 9

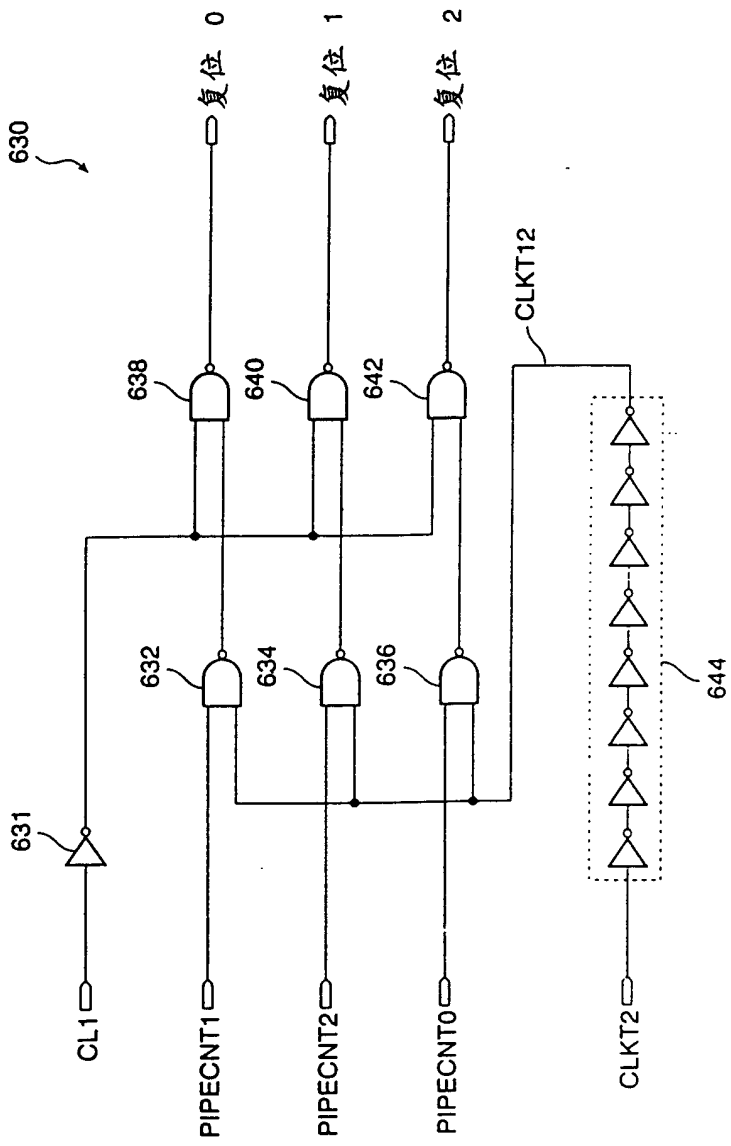


图 10

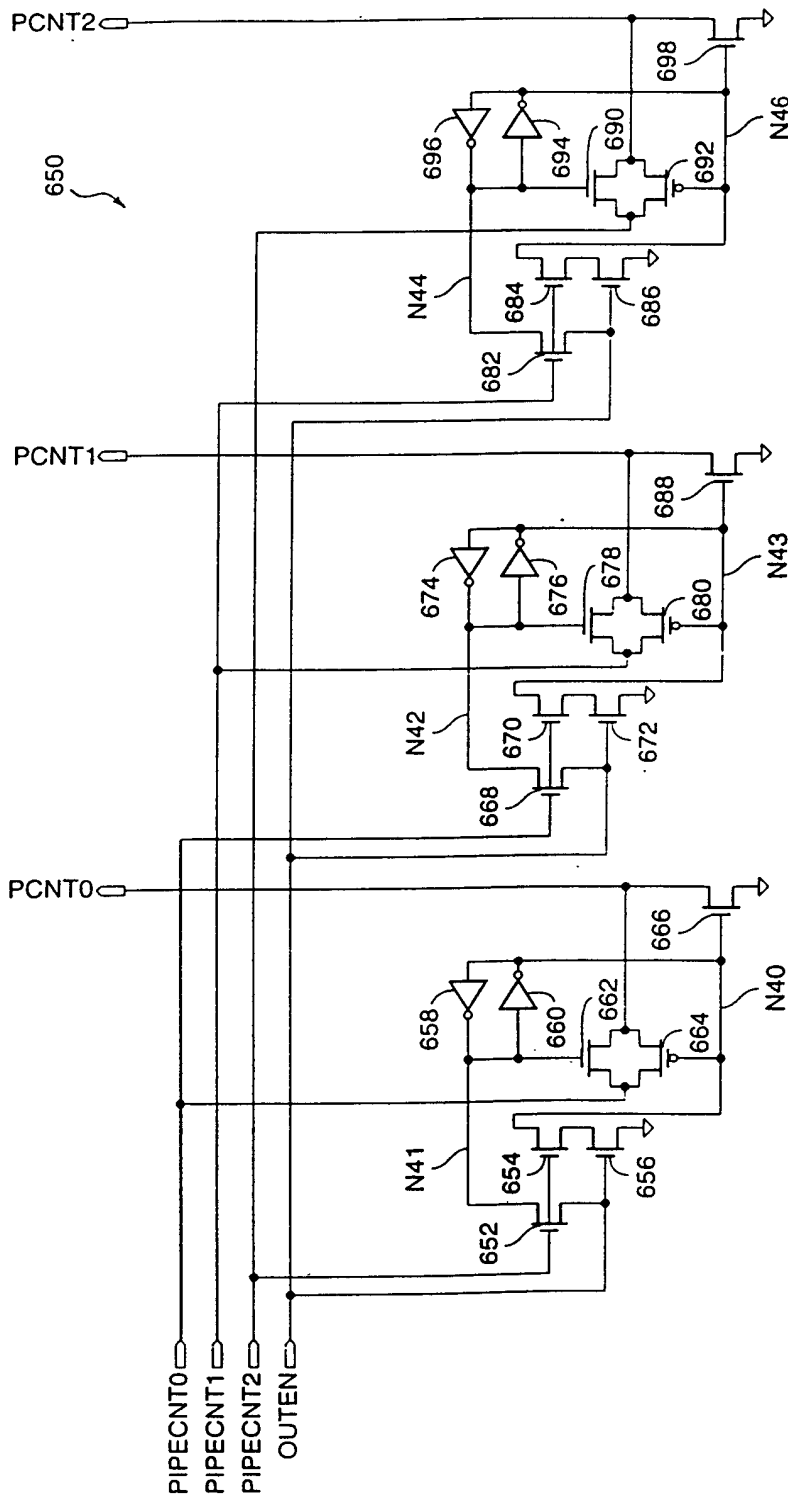


图 11