



(12) 发明专利

(10) 授权公告号 CN 101720465 B

(45) 授权公告日 2013. 10. 02

(21) 申请号 200880023085. 8

代理人 周建秋 王凤桐

(22) 申请日 2008. 05. 09

(51) Int. Cl.

(30) 优先权数据

60/917, 027 2007. 05. 09 US

60/917, 015 2007. 05. 09 US

60/917, 024 2007. 05. 09 US

60/917, 033 2007. 05. 09 US

12/117, 714 2008. 05. 08 US

G06F 17/50 (2006. 01)

G06F 1/12 (2006. 01)

(56) 对比文件

US 2004222857 A1, 2004. 11. 11, 摘要、说明书第 10-29 段、附图 1-4.

CN 1890754 A, 2007. 01. 03, 全文.

(85) PCT 申请进入国家阶段日

2009. 12. 31

审查员 徐卫锋

(86) PCT 申请的申请数据

PCT/US2008/006012 2008. 05. 09

(87) PCT 申请的公布数据

W02008/140791 EN 2008. 11. 20

(73) 专利权人 新思公司

地址 美国加利福尼亚州

(72) 发明人 R·C·迈克斯纳 M·拉罗歇

C·K·额 K·麦克尔文

(74) 专利代理机构 北京润平知识产权代理有限

公司 11283

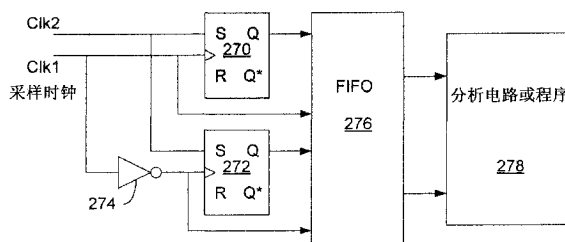
权利要求书4页 说明书25页 附图23页

(54) 发明名称

通过时钟采样进行相位关系计算

(57) 摘要

本发明公开了与用于自动化电路设计和模拟的时钟控制信号相关的各种技术。在某些实施例中,一种方法包括在第一时间接收具有第一相位关系的异步的第一时钟信号和第二时钟信号,并在第一时钟跳变时对所述第二时钟信号采样。所述方法还包括存储样本;以及分析该样本以确定所述第二时钟信号相对于所述第一时钟信号的第一相位关系并提供该第一相位关系的表示。还描述了其它实施例。



1. 一种用于制造设计模块的方法,该方法包括:

在第一时间,用计算机接收具有第一相位关系的第一异步时钟信号和第二异步时钟信号,其中所述第一时间是在所述第一异步时钟信号的第一跳变时;

在第一异步时钟信号的第一跳变时对第二异步时钟信号采样;

存储样本;以及

分析所存储的样本以确定所述第二异步时钟信号与所述第一异步时钟信号之间的所述第一时间的第一相位关系并提供该第一相位关系的表示,其中所述第一异步时钟信号和所述第二异步时钟信号通过耦合到配置成被测试的设计模块的电路而被接收,其中所述电路被配置成在触发事件发生之前基于所述第一相位关系来捕捉所述设计模块的状态。

2. 根据权利要求1所述的方法,该方法还包括在所述触发事件之前使用所述第一相位关系来确定所述设计模块的状态,并在所述触发事件之前基于所述设计模块的状态执行所述电路的模拟。

3. 根据权利要求1所述的方法,该方法还包括:

接收第三时钟信号,该第三时钟信号与所述第一异步时钟信号和所述第二异步时钟信号异步并具有与所述第一异步时钟信号的第一异步时钟信号的第二跳变时的第二相位关系;

在所述第一异步时钟信号的第二跳变时对所述第三时钟信号采样;

存储所述第三时钟信号的样本;

分析所述样本以确定所述第一时钟和第三时钟的所述第二相位关系并提供该第二相位关系的表示,其中所述第三异步时钟信号通过所述电路接收,其中所述电路还被配置成在所述触发事件发生之前基于所述第二相位关系捕捉所述设计模块的状态。

4. 根据权利要求1所述的方法,其中所述设计模块包括设计逻辑,其中所述电路还包括复制逻辑,该复制逻辑被配置成复制所述设计逻辑以提供初始条件信号,该初始条件信号指示所述触发事件之前所述设计逻辑的状态。

5. 根据权利要求1所述的方法,其中所述设计模块包括设计存储器,并且所述电路还包括设计寄存器和配置成复制所述设计存储器和设计寄存器以提供初始条件信号的复制存储器和复制寄存器,该初始条件信号指示在所述触发事件之前所述设计存储器和设计寄存器中至少一者的状态。

6. 根据权利要求1所述的方法,其中所述设计模块包括逻辑,且所述电路包括将到该逻辑的输入信号和延迟输入信号进行复用的复用器以及提供来自所述逻辑的输出信号和延迟输出信号以提供初始条件信号的解复用器,该初始条件信号指示所述触发事件之前所述逻辑的状态。

7. 一种用于模拟电路的方法,该方法包括:

用计算机生成电路的硬件描述,其中所述电路被配置成在第一时间接收具有第一相位关系的第一异步时钟信号和第二异步时钟信号,其中所述第一时间是在所述第一异步时钟信号的第一跳变时;

在第一异步时钟信号的第一跳变时对所述第二异步时钟信号采样;

存储样本;以及

用计算机分析所存储的样本以确定所述第二异步时钟信号与所述第一异步时钟信号

之间的所述第一时间的第一相位关系并提供该第一相位关系的表示,其中所述电路耦合到配置成被测试的设计模块,其中所述电路被配置成触发事件发生之前基于所述第一相位关系来捕捉所述设计模块的状态。

8. 根据权利要求7所述的方法,该方法还包括在所述触发事件之前使用所述第一相位关系来确定所述设计模块的状态,并在所述触发事件之前基于所述设计模块的状态执行所述电路的模拟。

9. 根据权利要求7所述的方法,其中所述设计模块包括设计逻辑,其中所述电路还包括复制逻辑,该复制逻辑被配置成复制所述设计逻辑以提供初始条件信号,该初始条件信号指示所述触发事件之前的所述设计逻辑的状态。

10. 一种用于制造设计模块的装置,该装置包括:

接收电路,该接收电路在第一时间接收具有第一相位关系的第一异步时钟信号和第二异步时钟信号,其中所述第一时间是在所述第一异步时钟信号的第一跳变时;

采样电路,该采样电路在第一异步时钟信号的第一跳变时对第二异步时钟信号采样;

存储电路,该存储电路存储样本;

分析逻辑,该分析逻辑分析所存储的样本以确定所述第二异步时钟信号与所述第一异步时钟信号之间的所述第一时间的第一相位关系并提供该第一相位关系的表示,以及

在触发事件发生之前基于所述第一相位关系捕捉被配置成被测试的设计模块的状态。

11. 根据权利要求10所述的装置,其中,所述接收电路和分析逻辑是单个集成电路芯片的一部分。

12. 根据权利要求10所述的装置,该装置还包括芯片接口和将所述第一相位关系提供给所述芯片接口的电路。

13. 根据权利要求10所述的装置,该装置还包括接收所述第一异步时钟信号和所述第二异步时钟信号的设计逻辑和复制设计逻辑,其中所述复制设计逻辑包括所述接收电路并提供初始条件信号。

14. 根据权利要求10所述的装置,该装置还包括接收所述第一异步时钟信号和所述第二异步时钟信号的设计存储器和设计寄存器及复制存储器和复制寄存器,其中所述复制存储器和复制寄存器包括所述接收电路并提供初始条件信号。

15. 根据权利要求10所述的装置,该装置还包括接收所述第一异步时钟信号和所述第二异步时钟信号的附加逻辑,并且该装置还包括电路,该电路包括将到所述逻辑的输入信号和延迟输入信号进行复用的复用器和提供来自所述附加逻辑的输出信号和延迟输出信号以提供初始条件信号的解复用器。

16. 一种用于制造设计模块的方法,该方法包括:

在第一时间,用计算机接收第一异步时钟信号和第二异步时钟信号,其中所述第一异步时钟和所述第二异步时钟在所述第一时间具有第一相位关系,其中所述第一时间是在所述第一异步时钟信号的第一跳变时;

根据所述第二异步时钟信号生成中间信号;

在所述第一异步时钟信号跳变时对所述中间信号采样;

处理所采样的中间信号;

存储经处理的所采样的中间信号的样本;以及

分析所存储的样本以确定所述第二异步时钟信号相对于所述第一异步时钟信号的所述第一时间相位关系并提供相位表示的表示,其中所述第一异步时钟信号和所述第二异步时钟信号通过耦合至配置成被测试的所述设计模块的电路而被接收,其中所述电路被配置成触发事件发生之前基于所述相位关系来捕捉所述设计模块的状态。

17. 根据权利要求 16 所述的方法,该方法还包括使用所述相位关系来确定在所述触发事件之前的所述设计模块的状态,以及所述触发事件之前基于所述设计模块的状态执行所述电路的模拟。

18. 根据权利要求 16 所述的方法,该方法还包括接收与所述第一异步时钟信号和所述第二异步时钟信号异步的第三时钟信号并以与所述第二异步时钟信号相同的方式来处理所述第三时钟信号,其中所述第三异步时钟信号通过所述电路被接收,其中所述电路还被配置成所述触发事件发生之前基于所述第二相位关系来捕捉所述设计模块的状态。

19. 根据权利要求 16 所述的方法,其中所述设计模块包括设计逻辑,其中所述电路还包括复制逻辑,该复制逻辑被配置成复制所述设计逻辑以提供初始条件信号,该初始条件信号指示所述触发事件之前的所述设计逻辑的状态。

20. 根据权利要求 16 所述的方法,其中所述设计模块包括设计存储器和设计寄存器,及其中所述电路还包括配置成复制所述设计存储器和设计寄存器以提供初始条件信号的复制存储器和复制寄存器,该初始条件信号指示所述触发事件之前所述设计存储器和设计寄存器中至少一者的状态。

21. 根据权利要求 16 所述的方法,其中所述设计模块包括逻辑,且所述电路还包括将到该逻辑的输入信号和延迟输入信号进行复用的复用器以及提供来自所述逻辑的输出信号和延迟输出信号以提供初始条件信号的解复用器,该初始条件信号指示所述触发事件之前的所述逻辑的状态。

22. 根据权利要求 16 所述的方法,其中所述中间信号表示由所述第一异步时钟信号进行的所述第二异步时钟信号的时钟跳变的检测。

23. 一种用于模拟电路的方法,该方法包括:

用计算机生成电路的硬件描述以在第一时间接收第一异步时钟信号和第二异步时钟信号,其中第一异步时钟和第二异步时钟在所述第一时间具有第一相位关系,其中所述第一时间是在所述第一异步时钟信号的第一跳变时;

所述电路根据第二异步时钟信号生成中间信号;

所述电路在第一异步时钟信号跳变时对所述中间信号采样;

所述电路处理所采样的中间信号;

所述电路存储经处理的所采样的中间信号的样本;以及

分析逻辑用计算机分析所存储的样本以确定所述第二异步时钟信号相对于所述第一异步时钟信号的所述第一时间相位关系并提供相位表示的表示,其中所述电路耦合至配置成被测试的设计模块,其中所述电路被配置成触发事件发生之前基于所述第一相位关系捕捉所述设计模块的状态。

24. 根据权利要求 23 所述的方法,其中所述设计模块的描述还包括对设计逻辑和配置成复制所述设计逻辑以提供初始条件信号的复制逻辑的描述,该初始条件信号指示所述触发事件之前所述设计逻辑的状态,并且其中所述第一异步时钟信号和所述第二异步时钟信

号由所述设计逻辑和复制设计逻辑来接收。

25. 根据权利要求 23 所述的方法,其中所述描述还包括对接收所述第一异步时钟信号和所述第二异步时钟信号的设计存储器和设计寄存器及复制存储器和复制寄存器的描述,其中所述复制存储器和复制寄存器被配置成复制所述设计存储器和设计寄存器以提供初始条件信号,该初始条件信号指示所述触发事件之前所述设计存储器和设计寄存器的状态。

26. 根据权利要求 23 所述的方法,其中所述描述还包括对接收所述第一异步时钟信号和所述第二异步时钟信号的逻辑的描述,以及对将到所述逻辑的输入信号和延迟输入信号进行复用的复用器和提供来自所述逻辑的输出信号和延迟输出信号以提供初始条件信号的解复用器的描述,该初始条件信号指示所述触发事件之前所述逻辑的状态。

27. 根据权利要求 23 所述的方法,其中所述中间信号表示由所述第一时钟信号进行的对所述第二时钟信号的时钟跳变的检测。

28. 一种用于制造设计模块的装置,该装置包括:

在第一时间接收第一异步时钟信号和第二异步时钟信号的电路,其中第一异步时钟和第二异步时钟在所述第一时间具有第一相位关系,其中所述第一时间是在所述第一异步时钟信号的第一跳变时;

根据第二异步时钟信号生成中间信号的电路;

在第一异步时钟信号跳变时对所述中间信号采样的电路;

处理所采样的中间信号的电路;

存储经处理的所采样的中间信号的样本的电路;以及

分析逻辑,该分析逻辑用于分析所存储的样本以确定所述第二异步时钟信号相对于所述第一异步时钟信号的所述第一时间的相位关系并提供相位表示的表示,其中所述装置耦合至配置成被测试的设计模块,其中所述装置被配置成触发事件发生之前基于所述相位关系捕捉所述设计模块的状态。

29. 根据权利要求 28 所述的装置,该装置还包括设计逻辑和用于提供初始条件信号的复制逻辑,并且其中所述第一异步时钟信号和所述第二异步时钟信号由所述设计逻辑和复制设计逻辑来接收。

30. 根据权利要求 28 所述的装置,该装置还包括接收所述第一异步时钟信号和所述第二异步时钟信号的设计存储器和设计寄存器及复制存储器和复制寄存器,其中所述复制存储器和复制寄存器提供初始条件信号。

31. 根据权利要求 28 所述的装置,该装置还包括对接收所述第一异步时钟信号和所述第二异步时钟信号的附加逻辑的描述,以及对将到所述逻辑的输入信号和延迟输入信号进行复用的复用器和提供来自所述附加逻辑的输出信号和延迟输出信号以提供初始条件信号的解复用器的描述。

32. 根据权利要求 28 所述的装置,其中所述中间信号表示由所述第一异步时钟信号进行的对所述第二异步时钟信号的时钟跳变的检测。

通过时钟采样进行相位关系计算

[0001] 相关申请

[0002] 本申请要求于 2007 年 5 月 9 日提交的美国临时申请 No. 60/917,015(档案号 No. 02986.P064Z)、于 2007 年 5 月 9 日提交的美国临时申请 No. 60/917,024(档案号 No. 02986.P066Z)、于 2007 年 5 月 9 日提交的美国临时申请 No. 60/917,027(档案号 No. 02986.P067Z) 以及于 2007 年 5 月 9 日提交的美国临时申请 No. 60/917,033(档案号 No. 02986.P068Z) 的权益;这些申请通过引用整体地结合到本文中。本申请与题为“Techniques for Use with Automated Circuit Design and Simulations”(档案号 No. 02986.P066) 的于 2008 年 5 月 8 日提交的申请 no. 12/117,705、题为“Techniques for Use with Automated Circuit Design and Simulations”(档案号 No. 02986.P066) 的于 2008 年 5 月 8 日提交的申请 no. 12/117,711 以及题为“Techniques for Use with Automated Circuit Design and Simulations”(档案号 No. 02986.P068) 的于 2008 年 5 月 8 日提交的申请 no. 12/117,693 在同一日提交。除本段、标题、发明内容、权利要求以及摘要之外,这四个申请是相同的。

技术领域

[0003] 本发明的实施例总体上涉及自动化电路设计和 / 或电路模拟。

背景技术

[0004] 集成电路(IC 或芯片)被用于包括例如个人计算机、嵌入式控制器、蜂窝电话、及其它通信设备的多种系统。电路设计者常常采用计算机辅助技术进行电路进行设计和模拟。已开发了诸如硬件描述语言(HDL)的标准语言来描述数字电路以辅助复杂数字电路的设计和模拟。包括但不限于 VHDL 和 Verilog 的多种硬件描述语言已逐渐成为行业标准。VHDL 和 Verilog 是通用硬件描述语言,其允许使用抽象数据类型来定义门级、寄存器传输级(RTL)或行为级的硬件模型。随着器件工艺继续进步,已开发了各种产品设计工具以使 HDL 适合用于新的器件和设计风格。

[0005] 在用 HDL 代码设计集成电路时,首先写代码并由 HDL 编译程序进行编译。HDL 源代码在某一层级描述电路元件,且编译程序由此编译产生 RTL 网表。RTL 网表通常是技术无关网表,因为其与特定供应商的集成电路(诸如,现场可编程门阵列(FPGA)或专用集成电路(ASIC)等)的技术/架构无关。RTL 网表对应于电路元件的示意性表示(与行为表示相反)。然后执行关联(映射)操作,以从技术无关 RTL 网表转换到可以用来产生供应商的技术/架构中的电路的技术专用网表。FPGA 供应商利用不同的技术/架构来实现其集成电路内的逻辑电路。因此,映射技术无关 RTL 网表以产生专用于特定供应商的技术/架构的网表。

[0006] 在编写并编译 HDL 代码之后,将集成电路(IC 或芯片)或包括多个 IC 的系统的设计验证至正确。不断进步的处理技术及设计尺寸和复杂性方面的迅速发展已导致对复杂电路(包括但不限于 ASIC)设计进行验证存在问题,这难以使用传统模拟工具和技术来解决。

当前的模拟工具的示例包括 ModelSim、VCS 和 NCSIM 模拟工具。

[0007] 结果,某些设计者使用诸如 FPGA 的多个 IC 来构造模型板。然而,仍然存在调试硬件设计的问题。当在调试期间检测到错误时,设计者可能尝试从电路分接(tap)感兴趣的信号,并使用逻辑分析器来确定错误的原因。JTAG(联合测试行动小组)是用于通过访问关于子块的信息来测试并调试集成电路的子块的著名技术。然而,这是艰难的过程且常常无效,尤其是在间歇性错误的情况下。可以使用模拟器来调试错误。然而,已经发生的错误常常难以重复和重构。此外,根据复杂性,模拟可能花费相当量的时间来到达已发生错误的点。

发明内容

[0008] 在某些实施例中,一种方法包括在第一时间接收具有第一相位关系的异步的第一时钟信号和第二时钟信号,并在第一时钟跳变时对所述第二时钟信号采样。所述方法还包括存储样本;以及分析该样本以确定所述第二时钟信号相对于所述第一时钟信号的第一相位关系并提供该第一相位关系的表示。

[0009] 在某些实施例中,一种方法包括在第一时间接收异步的第一时钟信号和第二时钟信号,其中第一时钟和第二时钟在所述第一时间具有第一相位关系。所述方法还包括根据所述第二时钟信号生成中间信号,在所述第一时钟信号跳变时对所述中间信号采样;以及处理所采样的中间信号。所述方法还包括存储经处理的所采样的中间信号的样本,并分析该样本以确定所述第二时钟信号在所述第一时间相对于所述第一时钟信号的相位关系并提供相位表示的表示。

[0010] 在某些实施例中,一种方法包括接收对包括设计模块的电路的描述,所述设计模块包括由具有第一频率的第一时钟信号来进行时钟控制的第一逻辑,该第一逻辑向由具有第二频率的第二时钟信号来进行时钟控制的同步器电路输出数据信号,所述同步器电路将所述数据信号提供给第二逻辑。所述方法还包括生成附加描述,该附加描述包括对复制模块和复制第二逻辑的描述,所述复制模块包括由所述第二时钟信号进行时钟控制的延迟电路,该延迟电路用于接收并传递从所述同步器电路输出的数据信号,所述复制第二逻辑用于接收被所述延迟电路输出的数据信号。

[0011] 如下所述,可以通过方法、装置、系统、芯片、软件、信号和运行软件或处理信号的计算机来实现其它实施例。

附图说明

[0012] 通过参考以下说明和用来示出本发明的实施例的附图,可以理解本发明的实施例。然而,本发明的实施例不限于这些图的细节。

[0013] 图 1 是根据本发明的某些实施例的包括计算机、接口以及芯片的系统的方框图表示。

[0014] 图 2 是根据本发明的某些实施例的包括计算机、制造设备、接口以及芯片的系统的方框图表示。

[0015] 图 3 是示出了根据本发明的某些实施例的生成用于配置或制造芯片的文件的程序的各种方面的流程图。

- [0016] 图 4 是示出了根据本发明的某些实施例的使芯片工作的各种方面的流程图。
- [0017] 图 5 是示出了根据本发明的某些实施例的在模拟器中使用初始条件信号和输入信号的各种方面的流程图。
- [0018] 图 6 是根据本发明的某些实施例的芯片中的具有相关电路的逻辑和复制逻辑 (replicated logic) 的方框图表示。
- [0019] 图 7 是根据本发明的某些实施例的芯片中的逻辑和相关电路及另一芯片中的复制逻辑和相关电路的方框图表示。
- [0020] 图 8 是根据本发明的某些实施例的具有相关电路的逻辑和复制逻辑的方框图表示。
- [0021] 图 9 是示出了本发明的某些实施例的各种方面的流程图。
- [0022] 图 10 是根据本发明的某些实施例的 RAM 硬件替代的方框图表示。
- [0023] 图 11 是根据本发明的某些实施例的用于确定异步时钟信号之间的相位关系的系统的方框图表示。
- [0024] 图 12 和图 13 均是图 11 的系统的时序图。
- [0025] 图 14 是根据本发明的某些实施例的用于确定异步时钟信号之间的相位关系的系统的方框图表示。
- [0026] 图 15 是图 14 的系统的时序图。
- [0027] 图 16 是根据本发明的某些实施例的用于确定异步时钟信号之间的相位关系的系统的方框图表示。
- [0028] 图 17 是图 16 的系统的时序图。
- [0029] 图 18- 图 20 均是根据本发明的某些实施例的设计模块和复制模块及相关电路的方框图表示。
- [0030] 图 21 是根据本发明的某些实施例的具有在快照 (snapshot) 解决方案中使用的电路的芯片的方框图表示。
- [0031] 图 22 是根据本发明的某些实施例的图 19 的芯片的一部分的示例的方框图表示。
- [0032] 图 23 是根据本发明的某些实施例的图 21 的芯片的一部分的示例的方框图表示。
- [0033] 图 24 是根据本发明的某些实施例的图 22 的电路的一部分的示例的方框图表示。
- [0034] 图 25 是根据本发明的某些实施例的图 23 的电路的一部分的示例的方框图表示。
- [0035] 图 26 是根据本发明的某些实施例的图 21 的芯片的一部分的示例的方框图表示。
- [0036] 图 27 是根据本发明的某些实施例的在快照解决方案中使用的电路的方框图表示。
- [0037] 图 28 是根据本发明的某些实施例的在时分复用器解决方案中使用的电路的方框图表示。
- [0038] 图 29 和图 30 均是根据本发明的某些实施例的用于图 28 的系统的时序图。
- [0039] 图 31 是根据本发明的某些实施例的在时分复用器解决方案中使用的电路的方框图表示。
- [0040] 图 32 是根据本发明的某些实施例的在时分复用器解决方案中使用的电路的方框图表示。
- [0041] 图 33 是图 32 的系统的时序图。

[0042] 图 34 示出了可以用于本发明的示例性计算机系统。

具体实施方式

[0043] 以下说明和附图是本发明的实施例的示例性说明且不应将其理解为限制本发明。描述了许多特定细节以提供对本发明的实施例的理解。然而,在某些情况下,未描述某些众所周知或常规细节以免使本发明含糊难懂。

[0044] 本文所述的本发明的某些实施例包括在发生错误或感兴趣的其它事件之前,对问题的逻辑设计的状态进行捕捉。本公开涵盖本发明的三个范畴:(1) 逻辑复制解决方案,(2) 快照解决方案,以及(3) 时分复用器解决方案。此外,存在可以用于本发明的任何范畴的多个发明特征(在第 III 节中描述)。本发明的某些实施例包括多于一个的解决方案和多于一个的所列附加特征。其它实施例包括用于对电路进行设计、从芯片提出初始条件信号和输入信号、并在模拟中使用该初始条件信号和输入信号的程序。本说明书中所使用的术语“芯片”包括“模块”。以下目录给出本公开的大纲。

[0045] I. 系统概述

[0046] II. 逻辑复制解决方案

[0047] III. 某些实施例的附加特征

[0048] A. 功能等效但不同的复制模块(replicated module)

[0049] B. 使初始条件与模拟器中的 HDL 代码相关联

[0050] C. 生成硬件替代列表

[0051] D. 访问内部状态

[0052] E. 形式验证流技术

[0053] F. 划分

[0054] G. 黑盒子

[0055] H. 多时钟域

[0056] I. 用较快时钟对较慢时钟采样

[0057] J. 用较快时钟对由较慢时钟生成的信号采样

[0058] K. 单独 FPGA 或 ASIC 中的复制

[0059] IV. 快照解决方案

[0060] V. 时分复用器解决方案

[0061] VI. 附加信息和实施例

[0062] 本公开使用短语“在某些实施例中”和“在其它实施例中”来指示本文所述的各种发明中的一个或多个的实现。对“某些实施例”的不同引用不一定指示相同的实施例。某些方法、装置、系统以及芯片可以包括多于一个的实施例。

[0063] I. 系统概述

[0064] 测试逻辑设计是通过运行模拟来分析故障的重要步骤。在模拟中可以使用导致实际错误的一组初始条件和输入来重新产生问题,以使得设计者能够最终找到消除该问题的解决方案。在示例性配置中,测试芯片直至发生故障。还通过接口来连接芯片以允许记录导致故障的输入和条件。该信息随后被馈送到模拟程序以分析问题的原因。

[0065] 参照图 1,芯片(或模块)14 通过接口 16(其可以是 JTAG 接口)与计算机 20 对

接。在某些实施例中,芯片 14 包括由计算机 20 中的程序通过 JTAG 接口 16 来配置的门阵列。该配置允许可通过 JTAG 接口看见芯片的各种信号。如本文详细描述的那样,在配置芯片 14 之后,使该芯片工作,以便收集其中发生错误或另一感兴趣事件的电路的初始条件信号(“初始条件”)和输入信号。本发明的某些实施例包括使输入信号延迟直至发生触发条件(例如错误条件),并随后冻结该延迟输入信号。某些复制寄存器或所添加的寄存器(以及某些实现方式中的其它存储器)的内容保持初始条件,所述内容为那些寄存器中被冻结的一组延迟输入信号的开头处或附近的(以及其它存储器)的内容。因此,所述延迟输入信号是导致触发条件的输入信号。所述初始条件和输入信号可以被提供给模拟器,以模拟(特别地)导致触发条件的感兴趣电路的工作。

[0066] 所述初始条件和输入信号通过 JTAG 接口 16 被提供给计算机 20 中的模拟器。通常,数据不是从接口被直接提供给模拟器,而是通过另一软件程序。所述模拟器包括在处理器 22 中运行的存储在存储器 24 中的软件程序。借助于初始条件和输入信号,所述模拟器可以详细地模拟芯片并尝试发现错误或其它感兴趣事件的性质。因此,可以将模拟器用于调试及其它目的。进行编程和配置的计算机不一定是进行模拟的计算机。其它实施例不包括门阵列且使用除 JTAG 之外的接口。

[0067] 为了分析故障,常常需要早期的数据传播,因为一旦发生故障,数据已显著地改变,从而掩蔽了故障的原因。此外,数据处理难以反向运行,因此,不能执行数据回溯以获取较早的数据。在示例性配置中,制造具有延迟输入的新芯片模块,从而提供错误的早期迹象,以允许逐步进行模拟。

[0068] 图 2 示出一种系统,其中计算机 30 包括用于向制造设备 40 提供芯片制造文件以制造芯片(或模块)38 的程序。图 1 的组件可以包括多于一个的计算机和制造设备的位于不同的位置并在不同的时间工作的各种部件。在制造芯片 38 之后,使该芯片 38 工作以收集与其中发生错误或另一感兴趣事件的电路相关的初始信号和输入信号。该初始条件和输入信号通过接口 42 被提供给计算机 30 中的模拟器。所述模拟器包括在处理器 32 中运行的存储在存储器 34 中的软件程序。借助于初始条件和输入信号,模拟器可以详细地模拟芯片,并尝试发现错误或其它感兴趣事件的性质。提供制造文件的计算机不一定是进行模拟的计算机。

[0069] 以下是与某些实施例相关的过程的概述。然而,在其它实施例中,细节可以不同于所列内容。设计者可以在源代码中选择一个或多个模块进行分析。可以选择某些信号以用于作用于调试硬件的触发器。这些信号的状态可以由设计者来指定,以定义用于触发调试硬件的精确条件。调试硬件被合成,生成网表,且最终设计完成逻辑配置(place and route)。各种软件可用于这些过程。例如,可以将合成工具用于硬件合成。然后将设计放置在诸如 FPGA 或 ASIC 的硬件器件中。运行硬件。一旦已检测到触发条件,则所有调试硬件连同用于该调试硬件中的延迟的先进先出电路(FIFO)一起被冻结。由控制计算机来获取 FIFO 和调试硬件的状态。此信息被变换成可以被软件模拟器使用的形式。所述设计被导入模拟器,并施加初始条件信号和输入信号。设计者或其他人员可以使用模拟器来调试硬件模块。在不同的实现方式中,存在不同程度的人为参与和程序实现的自动化。

[0070] 本文所讨论的电路可以用于多种类型的芯片,举例来说,包括微处理器、嵌入式控制器、芯片组、蜂窝电话芯片。

[0071] 图 3 是生成用于配置或制造芯片的文件的流程图。计算机 20 或 30 中的程序生成电路功能和布局或对电路的其它描述的文件（方框 50）。在某些情况下，所述描述可以包括比电路的功能和布局的完整描述少的内容。修订文件或生成新文件以包括对应于在本公开的第 I. -VI. 节中描述的一个或多个解决方案的电路（方框 52）。例如，程序对电路（被称为设计电路或原始电路）的描述进行分析，并使用在分析中获得的至少某些信息来生成用于电路的修订后的文件或新文件的附加描述。而且，除对设计电路的描述之外，还可以使用各种其它信息来生成新电路。描述的分析量取决于各种因素，包括多少信息被单独地提供给分析程序和程序必须为其本身收集多少信息。分析量还可以根据对程序给出的特定命令而变化。某些命令包括比其它命令更多的分析。分析的范围可以根据实现方式及其它情况而从大范围变为相对很小。

[0072] 可以由与在方框 50 中相同的程序或由不同的程序来得出修订后的文件。例如，修订后的文件可以指定被复制的电路组件。基于该修订后的文件而生成芯片配置或制造文件（方框 54）。然后，响应于该配置或制造文件来配置或制造芯片（方框 56）。

[0073] 存在各种用于表示本文所述的电路的技术。以下是简化概述。例如，可以编译电路的文本表示（例如以 HDL）以生成第一寄存器传输级（RTL）网表。RTL 网表可以示出寄存器及其它逻辑，该寄存器及其它逻辑被互连以通过在文本表示中描述的电路显示数据流。编译程序的示例是逻辑合成编译程序，其通常是在通用计算机系统上运行的计算机程序，虽然在某些实施例中，计算机系统可以是特定专用计算机系统。逻辑合成编译程序的示例是来自加利福尼亚州桑尼维尔的 Synplicity 公司的“Synplify®”程序。

[0074] 可以使第一 RTL 网表关联（映射）到诸如 FPGA 的目标架构，以生成用于调试电路的模型板。通常由集成电路的提供商来确定所述目标架构。目标架构的示例包括来自诸如 Altera、Xilinx、Lucent Technologies、先进微器件（AMD）以及莱迪思半导体等供应商的 FPGA 和复杂可编程逻辑器件。关联（映射）操作可以将期望电路的 RTL 级描述转换成使用目标架构的构造块实现的等效电路。可以生成技术专用网表。随后可以使用逻辑配置软件工具来生成目标架构中的电路的设计。

[0075] 调试过程可以识别电路中的有问题区域。设计者可以使用本文所述的解决方案，诸如选择复制电路的一部分以进行进一步分析。在本示例中，复制所选电路部分，并添加相关电路。可以重新编译该电路的文本表示以生成修订后的（例如第二）RTL 网表。可以使该修订后的 RTL 网表关联（映射）到目标架构，诸如 FPGA、ASIC 或诸如微处理器、通信芯片等的其它芯片。本发明不限于这些细节。可以使用各种其它程序。例如，复制电路的结合不必始终返回到源代码描述。复制电路的实现可以在 RTL 网表中进行，或者可以通过逐步修改或添加设计 RTL 网表而实现。

[0076] 图 4 是用于如本文所描述的那样使芯片工作以获得初始条件和输入信号的流程图。使芯片工作（方框 60）。响应于触发条件而生成触发信号（方框 62）。根据本公开的第 I. -VI. 节中所描述的一个或多个解决方案来获得初始条件和输入信号（方框 64）。所述初始条件和输入信号被提供给芯片接口（方框 66）。

[0077] 图 5 是用于使用初始条件和输入信号来执行模拟的流程图。已根据本公开的第 I. -VI. 节中所描述的一个或多个解决方案获得的初始条件和输入信号被接收以供模拟器使用（方框 70）。基于所述初始条件和输入信号来执行模拟（方框 72）。通过模拟来识别

关于与触发条件相关的电路工作的细节（方框 74）。模拟器可以使用各种测试或程序。

[0078] 图 3- 图 5 的流程图仅仅是某些实施例的概述。其它实施例可不遵循该流程图或包括其不同细节。在实际的实现方式中,可以使用各种附加程序和结构。

[0079] 本文所使用的术语“设计电路”、“设计存储器”和“设计寄存器”指的是在利用本公开所描述的技术之一对电路（或对附加电路的描述）进行分析和添加之前,作为初始设计（或对设计的描述）的一部分的电路、存储器和寄存器。设计电路可能被称为原始电路或起点处的电路。然而,设计电路可以包括众所周知的特征。如下文所解释的那样,电路（或对电路的描述）添加的示例包括逻辑复制和相关电路、快照寄存器和相关电路、以及复用器和解复用器及相关电路。

[0080] II. 逻辑复制解决方案

[0081] 用于获得初始条件信号和输入信号的逻辑复制解决方案包括对逻辑进行复制,以提供设计逻辑未提供的信息。图 8(类似于美国专利 6,904,574 的图 3) 示出了具有复制逻辑的系统。图 8 提供包括设计逻辑块 202 和复制逻辑块 204 的芯片中的电路的示例。例如,假设原始设计的调试表现出逻辑块 202 存在问题。因此,选择逻辑块 202,并复制作为复制逻辑块 204,以实现问题的进一步分析。例如使用图 3 的程序（方框 52）来生成复制逻辑 204。延迟逻辑 212(诸如 FIFO) 使提供给复制逻辑块 204 的输入信号延迟。延迟逻辑 212 可以包括诸如 FIFO 的典型电路逻辑和元件,其促使输入信号在比输入将到达设计逻辑块 202 的时间稍晚的时间到达复制逻辑块 204。这样,可以在看到设计逻辑块 202 中发生错误之后,对该错误进行分析,因为该错误将在稍后时间出现在复制逻辑块 204 中。

[0082] 触发逻辑 230 被插入电路中,以设置使电路的复制部分暂停的触发条件。还可以插入一个或多个控制器以控制触发逻辑。在本示例中,触发逻辑 230 具有两个输出:断点信号 218,其在被声明(asserted)时能够使得时钟控制逻辑 214 停止运行;以及延迟暂停信号,其在被声明时能够使得延迟逻辑 212 停止运行。所述延迟暂停信号还促使复制逻辑 204 将其寄存器和存储器内容冻结。

[0083] 时钟控制逻辑 214 控制提供给复制逻辑块 204 的时钟信号。时钟控制逻辑 214 可以包含允许提供给复制逻辑块 204 的时钟信号暂停以便在存在某些条件时使复制逻辑停止执行的典型逻辑和电路元件。时钟控制逻辑 214 还可以允许逐个时钟地单步遍历复制逻辑,以分析错误。可以设置断点 218 以便在存在某些条件时使时钟暂停,诸如在发生触发条件时。在某些实施例中,在正常工作期间,对电路进行时钟控制的系统时钟流过时钟控制逻辑 214 中的锁存器,并充当复制逻辑块 204 的时钟。断点信号可以将时钟切换到系统时钟的锁存版本,这可以由时钟控制信号来控制,以允许时钟信号暂停和逐个周期地单步递进。可以从复制逻辑 204 中的寄存器和存储器获得初始条件信号,且可以从延迟逻辑 212 获得输入信号。本发明不需要包括所示和所述的细节,而是可以以其它方式来实现。

[0084] 在某些实施例中,本发明公开了实现逻辑复制解决方案的各种创新方式。在一种实施例中,可以提供附加串行电路以便连续地从复制模块接收数据。例如,图 6 包括芯片 100,芯片 100 包括设计电路 104 和复制模块 106。设计电路 104 包括被测试设计逻辑 110 和附加逻辑 112。被测试设计逻辑 110 可以被称为设计模块。被测试设计逻辑 110 和附加逻辑 112 是意图在感兴趣芯片的生产版本中使用的电路。当然,在设计的生产版本完成之前,可以修改设计电路。在本示例中,被测试设计逻辑 110 包括寄存器 116 和 118,且可以包

括多种其中组件,诸如附加寄存器、存储器、各种电路以及某些实施例中的固件。复制模块 106 包括被测试复制逻辑 130,该被测试复制逻辑 130 包括类似于被测试设计逻辑 110 中的电路,包括类似于寄存器 116 和 118 的寄存器 138 和 140。

[0085] 在工作中,设计逻辑 110 和复制逻辑 130 均接收相同的输入,但复制逻辑 130 接收输入的延迟版本。例如,设计逻辑 110 从导体 114 接收输入信号(包括数据输入),并从附加逻辑 112 接收附加输入信号(包括信号中的附加数据)。复制逻辑 130 通过延迟电路 134 从导体 114 接收输入信号,并通过延迟电路 136 从附加逻辑 112 接收附加输入信号。在图 6 的示例中,延迟电路 134 和 136 是 FIFO,该 FIFO 具有等于输入信号的宽度的宽度,以及等于用于给定类型的情况和/或信令的期望延迟的深度。由来自导体 126 的一个或多个时钟信号对设计逻辑 110、设计逻辑 130、FIFO 134 以及 FIFO 136 进行时钟控制。虽然只由一个时钟信号来对 FIFO 134 和 136 进行时钟控制,但也可以由多于一个的时钟信号来对逻辑 110 和 130 进行时钟控制。可以存在由一个或多个其它时钟信号来进行时钟控制的图 6 未示出的附加 FIFO。

[0086] 所述输入信号可以包括不同类型的信号,包括数据、地址、控制以及其它信号。有时,这些输入信号仅被称为数据信号,虽然它们可以不仅仅包括传统数据。设计和复制模块可以包括存储器和寄存器。

[0087] 例如,使用图 3 的程序(方框 52)来设计复制模块 106。程序的不同实现方式具有不同程度的设计者参与。例如,在某些实现方式中,设计者下命令,且程序生成复制模块。在其它实现方式中,设计者给出关于复制模块的更多细节,并且在某些实现方式中,设计者可以交互地参与复制模块的设计。

[0088] 在工作中,触发逻辑 122 监视设计逻辑 110。当发生被称为触发条件的特定感兴趣事件时,触发逻辑 122 向 FIFO 134 和 136 提供使 FIFO 停止响应时钟信号的触发信号。该触发信号还前进到复制逻辑 130,以冻结寄存器 138 和 140 中的寄存器值。触发逻辑 122 还向控制电路 124 提供促使控制电路 124 进入控制模式的信号,在控制模式下,使得 FIFO 134 和 136 的内容可作为用于模拟器的输入信号,且使得复制逻辑 130 的某些内容可作为用于模拟器的初始条件。可以将寄存器 138 和 140 链接在一起以提供初始条件。图 6 示出了其中也受到触发逻辑的影响的寄存器 138 和 140 在控制逻辑 124 的控制下通过复用器 144、寄存器 146、复用器 148 以及寄存器 150 来提供初始条件信号的示例。复用器 144 还可以接收 JTAG 链的输入。控制电路 124 可以是 JTAG 控制电路。控制电路 124 或其它电路可以考虑 FIFO 134 和 136 的输出与寄存器 146 和 150 的内容之间的任何时间延迟。复用器 144、148 和寄存器 146、150 的电路组合仅表示在芯片之外获得初始条件的特定方式。可以使用其它电路,诸如用于扫描并获取复制逻辑 130 中的寄存器值的串行扫描电路。

[0089] 在一种实施例中,复制模块可以位于相同或不同芯片中。图 7 示出了设计电路 104、触发逻辑 122 和控制电路 124 可以在芯片 160 中,而复制模块 106 可以在芯片 162。将复制模块放在不同的芯片中的原因是在芯片 160 中可能没有用于该复制模块的足够空间。

[0090] III. 某些实施例的附加特征

[0091] 以下是某些附加特征,除非另外指示,否则其可以与三个主要解决方案(在第 II、IV 和 V 节中)中的每一个的某些实施例相结合地使用。然而,每个主要解决方案中的其它实施例不包括这些特征中的一个、某些或全部。此外,还可以脱离主要解决方案而使用这些

附加特征中的某些特征。

[0092] A. 功能等效但与设计电路不同的复制电路

[0093] 在一种优选实施例中,复制模块保留设计模块的功能,但具有与设计模块不同的用于促进调试或用于简化复制电路的不同电路。复制电路可以在收集条件或输入以帮助分析潜在故障的焦点方面不同于设计电路。

[0094] 复制模块可以具有电路修改方面的自由以促进调试。例如,在复制模块中可以抑制电路优化。复制模块的目的是提供用于调试模拟的初始条件和输入,因此,实际上可能出于调试的目的,需要在效率方面看起来无关的电路。一方面,在复制模块中执行硬件替代。例如,为了便于调试,可以用复制模块中的普通电路来代替设计模块中的专有电路。提供对未暴露状态的访问可能是困难的任务,硬件替代在提供对未暴露状态的访问以获得设计模块中的黑盒子(black box)的内部状态时可能特别有用。对诸如内部状态的未暴露状态的访问可以包括提供控制电路,以获取诸如存储器中的存储数据等模块内部状态。在另一方面,可以将设计模块划分以便复制模块只包含设计模块的一部分,或者省略设计模块的某些部分。所述划分还可以包括从设计模块内部提供给复制模块的多个延迟电路,例如以便解决各种时序问题。

[0095] B. 使初始条件与模拟器中的 HDL 代码相关联

[0096] 某些实施例包括使来自芯片的至少某些初始条件的值与模拟器中的 HDL 代码内的特定区段(location)相关联(亦即映射或关联)。所述特定区段可以是 HDL 中的对象,诸如信号、寄存器、导体、存储器等。在某些实施例中,这包括阻止用来将 HDL 设计关联到网表中的编译程序软件内的顺序优化。应避免的顺序优化的示例包括但不限于重新定时、管线化(pipelining)、布置复制(placement replication)和状态分配。在关联中使用的编译程序软件可以按照某一次序生成模块中的所有存储器和储存元件的列表,所述次序可以与模拟器处理器获得初始条件值的顺序相关联(映射或相关)。存储元件的名称可以被直接关联到可与软件模拟器兼容的信号实例。

[0097] 与图 1、图 6 和图 9 相结合地提供了一种示例。图 1 中的计算机 20 内的一个或多个程序生成用于生成图 6 的电路的电路功能和布局的文件(包括 HDL 代码形式的文件)(图 9 的方框 240)。计算机 20 内的同一或另一程序修改并优化功能和布局设计(图 9 的方框 242)。本文所使用的术语“优化”仅仅意指改善,而不一定达到数学最佳效果。配置或制造包括图 6 的电路的芯片(图 9 的方框 244)。如结合图 6 所述那样,响应于触发信号,收集初始条件和输入信号,并使其可用于可以在计算机 20 或另一计算机上运行的模拟程序。使用关联技术来使所收集的初始条件与模拟器中的 HDL 代码内的特定区段相关联(图 9 的方框 246(1))。例如,所述关联可以是用于 FPGA 的关联(映射)网表与 HDL 源代码之间的关联。当然,在图 9 的方框之间可以存在各种中间文件和修订后的文件。

[0098] 在关联技术中可以使用各种试探法。一种此类试探法是电路组件的命名。可以建立扫描链中的初始条件状态的排序与 HDL 代码中所建立的数据相关元件的名称和区段之间的联系。源代码可以被用于所复制的所有逻辑。在黑盒子的情况下,例如没有内部器件的详细信息的路径块,可以将其输出作为到复制设计中的附加输入来处理。来自原始设计中的黑盒子的输出可以穿过 FIFO 到达复制电路中的那些输入。合成工具可以支持基于每个模块来禁用重新定时和顺序优化。其它实施例不包括本节的细节或包括其各种变体。

[0099] 某些实施例提供一种通过抑制用来将 HDL 设计映射到网表中的编译程序软件内的顺序优化而将从硬件收集的初始条件的值与 HDL 代码内的特定区段相关联的方法。在映射中使用的编译程序软件将按照某一次序创建模块中的所有存储元件的列表,所述次序可以与处理器获得初始条件值的顺序相关联。存储元件的名称可以被直接关联到与软件模拟器兼容的信号实例。

[0100] C. 生成硬件替代列表

[0101] 某些实施例包括通过生成所有硬件替代的列表(该替代由编译程序在生成网表时执行)来关联从硬件收集的初始条件的值。该替代列表用来从物理实现(physical implementation)到与软件模拟器兼容的 HDL 级表示追溯信号名称和值。在一种实施例中,可以用等效电路来取代专有模块。所述等效电路可以例如通过提供对未暴露状态的访问来促进调试。

[0102] D. 访问内部状态

[0103] 当要复制存储元件时,精确的复制品可能不可用于调试目的,因为原始存储元件通常不访问内部状态。也就是说,原始存储器具有某些未暴露状态。复制存储器通常将提供对那些状态的访问。复制存储器可以是替代存储器(硬件替代的一种示例),因为其包括允许访问未暴露状态的附加结构。复制存储器可以包括允许访问存储器数据的控制电路。提供关联(映射)以关联模拟器中的 HDL 代码中的替代存储器。图 9 的流程图可以特别地应用于方框 246 的部分(2)。

[0104] 图 10 示出了替代存储器。复制随机存取存储器(RAM) 254 可以与设计(原始)RAM 相同。控制电路 252 在触发条件之后冻结 RAM 254,并且还允许通过扫描链 256 对 RAM 254 的内容进行时钟输出(clock out),以便例如向外提供串行数据至控制计算机。来自复制模块的地址和来自复制模块的数据被提供给控制电路 252 和 RAM 254。在某些实施例中,必要时,合成工具可以调用例如 C 库中的函数,该函数将返回指向包含替代 RAM 设计的网表的指针。

[0105] 某些实施例包括电路和方法,该电路和方法通过替代包含诸如图 10 中的控制电路 252 等访问控制逻辑的专用 RAM 结构来处理复制逻辑中的各种形式的 RAM。所述访问控制逻辑允许专用 RAM 结构在所有正常读/写信号和时钟被启用时像正常 RAM 一样工作。访问控制逻辑允许一旦检测到触发条件,则冻结 RAM 的工作。然后,访问控制逻辑允许将 RAM 的内容扫描输出到控制计算机。访问控制逻辑还可以包括允许 RAM 的内容的串行输出的扫描链。该 RAM 内容被用来收集 RAM 的初始条件,以用于软件模拟。其它实施例不包括本节的细节或者包括其变体。例如,还可以将 RAM 视作黑盒子。RAM 的输出被馈入到延迟 FIFO,该延迟 FIFO 向输入模块馈送类似于输入的项目。在不能推断 RAM 结构的情况下,例如 RAM 块(BlockRAM),这种方法是优选的。

[0106] 某些实施例包括处理复制逻辑中的各种基本专有逻辑模块(例如,XilinxSRL16 移位寄存器和 Altera Altshifftap 移位寄存器)的电路和方法。这包括用允许访问那些模块的内部状态的逻辑等效电路来替代这些专有模块。可以由计算机来扫描这些内部状态,并在软件模拟期间将其用作初始条件。例如,可以用具有可访问内部状态的等效电路来代替 Xilinx SRL16 移位寄存器(该移位寄存器表示具有用于连接到外部电路的输入和输出引脚的专有模块)。其它实施例不包括本节的细节或者包括其变体。

[0107] E. 形式验证流技术

[0108] 过去,已使用形式验证流技术来确认实际设计与 HDL 代码中的描述匹配。某些实施例包括通过使用形式验证流技术来关联来自硬件的初始条件的值。在图 9 中特别地通过方框 246 (3) 提供了一种示例。其它实施例不包括此特征或者包括其变体。

[0109] F. 划分

[0110] 在一种实施例中,复制模块包括从设计模块内部到复制模块的各个位置的附加延迟电路。例如,第一延迟电路将设计模块的输入连接到复制模块的输入。第二(和可选的其它)延迟电路从设计模块的任何位置连接到复制模块中的适当位置。附加延迟电路可以提供模块之间的精确时序,尤其是对于多时钟域。

[0111] 在某些实施例中,设计模块被划分成多个设计段。一方面,所述设计段可以被全部复制,每个复制段通过延迟电路连接到其各自的设计段。另一方面,只复制设计段的一部分,每个复制段通过延迟电路连接到其各自的设计段。在设计模块中,部分设计段可以被相互分离或嵌入。另一方面,不对设计段的至少一部分进行复制,并且被复制的每个设计段向延迟电路提供输出信号,其中延迟电路的输出被提供给复制段各自的位置。

[0112] G. 黑盒子

[0113] 设计模块可以被划分成段,且其中至少一个段(例如,黑盒子模块)未被复制。电路设计常常包括黑盒子,例如具有未知内部状态或行为的模块,诸如 IP 块或专用逻辑模块。本发明的某些实施例包括在不需要详细内部电路的情况下处理黑盒子和各种专有逻辑模块的电路和方法。在某些实施例中,具有未知行为的专有逻辑未被拷贝到复制逻辑中。作为替代,这种方法将设计模块中的专有模块的输出作为到复制逻辑内的其各自位置的附加输入来处理。例如,图 20 包括设计模块 360 和复制模块 362。设计模块包括逻辑 364,逻辑 364 的输出被提供给黑盒子 366。黑盒子逻辑 366 为执行未知的各种功能的电路,虽然输入和输出信号是已知的。黑盒子逻辑 366 的输出被提供给设计模块 360 中的逻辑 368,并且还提供给复制模块 362 中的延迟电路 FIFO 376 以负责(account for)总的复制逻辑延迟。FIFO 376 的输出被提供给逻辑 378。在这方面,黑盒子逻辑 366 未在复制模块 362 中被直接复制。在图 20 的示例中,逻辑 374 的输出未被提供给逻辑 378。在某些实施例中,逻辑 374 未被复制,且 FIFO 376 的输出被连接到其在复制模块内的各个输入位置。响应于触发条件,FIFO 376 的内容被计算机获得,并用来在软件模拟中模拟专有逻辑 366 的输出。在一种实施例中,提供了延迟电路(未示出)以便将逻辑 364 的输入与复制逻辑 374 的输入相连。其它实施例不包括本节的细节或者包括其变体。

[0114] H. 多时钟域

[0115] 可以基于时钟域(每个时钟域接收单独的延迟输入)来划分设计模块。某些实施例包括处理由两个离散异步时钟进行时钟控制的两个逻辑电路之间的数据传输时序的电路和方法。此电路和方法可以包括使用“同步触发器(synchronizer flip-flop)”来对 FIFO 进行馈送,该 FIFO 的深度由两个时钟的相对频率来确定。图 18 示出设计模块 302 中的同步触发器 312。所述电路和方法可以被扩展至用来使交织于(crossing)任何数目的时钟域之间的一个或多个信号同步。

[0116] 许多设计使用多个独立时钟。当信号通路从一个时钟域跨越到另一个时钟域时,会出现时序问题。图 18 示出了信号从时钟 1 域转到时钟 2 域的示例。在设计模块 302 中,

由时钟 1 进行时钟控制的触发器 306 接收到数据信号。触发器 312 的输出被提供给逻辑 308, 逻辑 308 是时钟 1 域的一部分。当然, 可以并行地存在更多的数据信号。因此, 触发器 306 和触发器 312 均表示多个并行触发器。逻辑 308 的输出被提供给由时钟 2 进行时钟控制的同步触发器 312。触发器 312 的输出被提供给作为时钟 2 域的一部分的逻辑 314。

[0117] 复制模块 304 也接收时钟 1 和 2, 以便到复制模块的数据准确地仿真设计模块中的数据。复制模块 304 包括接收数据并由时钟 1 进行时钟控制的触发器 326。该数据可以是被触发器 306 接收到的数据且可以被也可以不被延迟电路 (未示出) 延迟。触发器 326 的输出被提供给作为时钟 1 域的一部分的逻辑 328。逻辑 328 的输出被提供给由时钟 2 进行时钟控制的同步触发器 332。由合成工具来识别穿过时钟域边界的所有信号。

[0118] 在图 18 的示例中, 同步触发器 332 的输出未被传递到作为时钟 2 域的一部分的逻辑 334 (该逻辑 334 由时钟 2 进行时钟控制)。然而, 在某些实施例中, 触发器 332 的输出被用来收集电路的状态, 例如作为寄存器 312 的内容的指示。在某些实施例中, 不包括同步触发器 332。

[0119] 穿过时钟域的信号使用设计模块 302 中的同步器 312。触发器 312 的输出在其被由时钟 2 进行时钟控制的 FIFO 320 延迟之后, 提供给逻辑 334。这保证逻辑 334 将得到与逻辑 314 相同的数据。响应于触发条件, FIFO 320 被冻结, 以便不接受新数据且其内容被扫描作为扫描链的一部分, 以获得用于模拟器的输入信号。可以从逻辑 334 中的寄存器和存储器获得初始条件信号。

[0120] 可以由与数据被时钟输入到 FIFO 中的相同时钟来对 FIFO 进行时钟控制。对于一个时钟域来说, FIFO 的深度可以不同于另一域中的另一 FIFO。可以计算该深度以便对所有域中的所有时钟提供相同的延迟时间量。一旦检测到触发条件, 则可以冻结用于在时钟域之间载送信号的 FIFO。FIFO 的内容可以被时钟输出到扫描链以供在模拟中使用。

[0121] 当设计中存在多个时钟时, 可以遵循特定方针以保证域之间的时序是正确的。例如, 在图 19 中, 数据信号通过时钟 1 信号 (时钟 1) 时钟而被输入到设计模块 340 中。在设计模块 340 和复制模块 348 内部, 存在时钟 2 的时钟域流失 (running off)。数据信号还通过深度 N1 的输入 FIFO 344 被提供给复制模块 348。来自设计模块 340 的数据通过深度 N2 的域间 FIFO 346 被提供给复制模块 348。复制模块 346 的总延迟可以取决于时钟 1 的周期和输入 FIFO 346 的深度 N1。通常, 延迟电路的总延迟与时钟的周期成反比。域间 FIFO 346 的目标可以是提供如下所述的延迟。假设时钟 2 的周期是已知的, 则可以如下所述这样计算深度 N2。如果 P1 是时钟 1 的周期且 P2 是时钟 2 的周期, 则可导出 N2:

[0122] $(P1 \times N1) / (P2 \times N2) = 1$ 或如在实践中一样接近于 1。

[0123] 如果 $P1 \times N2$ 能被 $P2$ 整除, 则:

[0124] $N2 = (P1 \times N1) / P2$ 。

[0125] 否则, N2 可以是下式的模数部分:

[0126] $N2 = ((P1 \times N1) / P2) + 1$ 。

[0127] N2 可以具有足够的延迟量以补偿由输入 FIFO 引入的延迟。然而, 仍存在不确定性, 即来自 FIFO 346 的输出的数据是什么样的。根据在模拟开始时对两个时钟进行的相对校准, 可能需要再一次对时钟 2 域中的复制硬件进行时钟控制。如果满足以下标准, 则对于复制模块 348 需要再一次对时钟 2 进行时钟控制。使 Y 是数字:

[0128] $Y = ((P1 \times N1) / P2) + 1$

[0129] 使 T 表示时间段：

[0130] $T = \text{fract}(Y) \times P2,$

[0131] 其中 $\text{fract}(Y)$ 是 Y 的小数部分。然后, 给定延迟 D, 其中 $T < D < P2$, 可能需要对硬件时钟 2 进行时钟控制再多一个周期。否则, 每当 D 落在 0 与 T 之间时, 可能不需要进行附加的时钟控制。

[0132] 可能不能确定时钟 1 和 2 的活动边沿 (active edge) 之间的关系, 直至已完成捕捉之后。因此, 可以遵循以下顺序: 过程开始; 触发器条件发生; 确定时钟的相位关系; 如果满足用于时钟 2 的延迟条件, 则发送时钟 2 的脉冲; 以及收集寄存器和 RAM 的状态。其它实施例不包括本节的细节或者包括其变体。

[0133] I. 用较快时钟对较慢时钟采样

[0134] 某些实施例包括通过直接在设计中用较快时钟对较慢时钟采样来计算两个离散异步时钟之间的关系的电路和方法。如果每个时钟的频率和占空因数是已知的, 则可以通过调整较慢时钟的偏移以与采样数据模式相匹配来计算一个时钟与另一个时钟之间的关系。此过程可以用来确定两个或更多时钟之间的关系。将每个较慢时钟信号与最快 (最高频率) 时钟信号相比较。请注意, 虽然时钟一般是异步的, 但在特定情况下, 可以证明两个或更多时钟是同步的 - 但是所述电路和方法对此并无要求。

[0135] 在 multi-clock 设计中, 每个时钟的频率是已知的。合成工具可以为每个域提供此信息。活动时钟边沿相互之间的关系在数据捕捉开始时是未知的。例如, 在图 6 中, 假设在导体 126 上向设计逻辑 110 和复制逻辑 130 提供图 12 的时钟信号 (时钟) 1 和时钟信号 (时钟) 2。在时间 t_0 , 当导体 126 和 114 上的时钟和数据被提供给设计逻辑 110 时, 时钟 1 和 2 具有特定的相位关系。表示此相位关系的一种方式是在时钟 1 的上升沿与时钟 2 的随后的下一个上升沿之间的时间。在时间 $t_0 + \text{延迟}$, 导体 114 上的数据通过延迟电路 134 到达复制逻辑 130。在时间 $t_0 + \text{延迟}$, 导体 126 上的时钟 1 和 2 之间的相位关系通常将与在时间 t_0 时不同。

[0136] 可以通过用设计中的最快时钟对那些时钟采样来分析每个时钟的时钟边沿, 从而估计所述相位关系。时钟样本可以被存储在 FIFO 中, 该 FIFO 可以被扫描, 并由诸如电路、模拟器或其它软件等分析逻辑或模块来分析。在大多数情况下, 增加样本的数目将减小边沿周围的模糊窗口 (window of ambiguity)。可以以不同的方式来处理对时钟的采样。在所有时钟均具有例如至少约为 50% 的占空因数的情况下, 由最快时钟直接对经采样的时钟进行采样, 且结果被放入 FIFO 中。为了提高准确度, 可以在最快时钟的上升沿和下降沿两者上对时钟采样。

[0137] 例如, 在图 11 中, 触发器 270 和 272 接收时钟 2 作为输入。由时钟 1 来对触发器 270 进行时钟控制且由通过反相器 274 的反相时钟 1 来对触发器 272 进行时钟控制, 以便触发器 270 在时钟 1 的上升沿上对时钟 2 采样且触发器 272 在时钟 1 的下降沿上对时钟 2 采样。触发器 270 和 272 中的采样值被时钟输入到 FIFO 276 中, FIFO 276 可以被扫描, 并由诸如电路或程序 278 等分析逻辑或模块来分析以确定时钟布置 (clock placement)。在 FIFO 276 与电路或程序 278 之间可以存在附加电路。

[0138] 图 12 提供时钟 1 和 2 的示例, 且时钟 2 通过时钟 1 的上升沿和下降沿而被采样。

本权利要求的电路和方法包括多于两个的时钟。例如,可以由时钟 1 以与时钟 2 相同的方式对类似于时钟 2、但与时钟 1 和 2 不同相(并且可能还处于不同频率)的时钟 3 进行采样。如上文所解释的那样来确定第一和第二时钟信号的相位关系,该关系是至少一个特定时间(例如第一时间)。针对至少一个特定时间(例如可以与第一时间相同或不同的第二时间)来确定第一和第三时钟信号的相位关系。其它实施例不包括本节的细节或者包括其变体。

[0139] J. 用较快时钟对根据较慢时钟生成的信号采样

[0140] 某些实施例包括通过在较快时钟的采样周期内检测较慢时钟的每个边沿跳变(edge transition)来计算两个离散异步时钟之间的关系的方法和电路。如果每个时钟的频率和占空因数是已知的,则可以通过调整较慢时钟的偏移以与所检测的边沿数据模式相匹配来计算一个时钟与另一个时钟之间的关系。此电路和方法可以用来确定两个或更多时钟之间的关系。

[0141] 处理时钟的此第二种方式可适用于将被采样的时钟具有例如基本小于 50% 的占空因数的情况。如果占空因数过小,则图 11 的采样器可能不能发现某些样本、乃至单个样本。例如,图 13 示出了具有远远小于 50% 的占空因数的时钟 1 和时钟 2。如果在图 11 的电路中由时钟 1 来对时钟 2 采样,则结果可能如图 13 所示,其中输出是恒定的,因为时钟 2 的占空因数太窄以至于不能被时钟 1 捕捉。没有捕捉到可以用来校准时钟的有效数据。

[0142] 与图 11 的电路相反,作为直接对时钟采样的替代,可以使用时钟来驱动例如图 14 所示的二次分频电路。这得到只有在检测到活动边沿时才改变的信号。参照图 14,由时钟 2(Clk2)进行时钟控制的触发器 282 的负(Q*)输出被反馈到其输入。触发器 282 的正(Q)输出作为输入被提供给由时钟 1 进行时钟控制的触发器 284。触发器 284 的输出作为输入被提供给也由时钟 1 进行时钟控制的触发器 286 和异-或(XOR)门 288。触发器 286 的 Q 输出也被提供给 XOR 门 288。XOR 288 的输出被提供给 FIFO 以供稍后分析。当在时钟 1 的采样周期期间出现活动时钟 2 边沿时,XOR 门的输出为高。这产生如图 15 所示的结果,该图显示了时钟 2 的每个活动边沿映射到由时钟 1 域进行时钟控制的的正电平输出。因此,图 14 的电路与时钟 2 的占空因数无关。可以存在与时钟 1(最高频率时钟)相比较的一个或多个附加时钟信号。该附加时钟信号可以是异相的,并可能具有与其它时钟不同的频率。图 14 可以被修改为包括附加触发器,以便电路对上升时钟边沿和下降时钟边沿进行响应。某些芯片可以具有多个采样器,某些与图 11 中的一样而某些与图 14 中的一样。其它实施例不包括本节的细节或者包括其变体。

[0143] 图 16 和图 17 示出了图 14 和图 15 的可选方式。参照图 16 和图 17,双时钟(双 Clk1)是时钟 1(c1k1)的上升沿和下降沿上的脉冲。图 16 类似于图 14,不同的是双 Clk1 被提供给触发器 284 和 286 的时钟输入端。结果得到的时序图在图 17 中示出,其中通过使用双 Clk1 进行 Clk2 正沿检测。还可以使用负沿检测。在电路的不同实施例中(诸如图 11、图 14 和图 16 的那些及本公开所示的其它电路),可以修改电路,从而使得根据实施例在时钟信号的上升跳变处、在时钟信号的下降跳变处、或在时钟信号的上升和下降跳变处进行采样。也就是说,在某些实施例中,在上升跳变处进行采样;在其它实施例中,在下降跳变处进行采样;并且在另外的其它实施例中,在上升和下降跳变处进行采样。

[0144] K. 单独 FPGA 或 ASIC 中的复制

[0145] 某些实施例包括在单独的 FPGA 或 ASIC 器件中实现部分逻辑的复制的方法（诸如图 3 中的方法）。例如，在图 6 中，复制模块 130 与设计逻辑 110 在同一芯片中，而在图 7 中，复制模块 130 与设计模块 110 在不同的芯片中。本方法不仅仅适用于从一个芯片到另一芯片的复制，而且可以应用于从多个芯片到多个附加芯片的复制（例如 2 个芯片到 4 个芯片、3 个芯片到 6 个芯片等）。在其它实施例中，多个芯片中的电路的复制部分可以被放置在单个芯片中或数目比原始芯片少的芯片中。例如，2 个芯片中的电路的复制部分可以均被放置在一个附加芯片中，或者 4 个芯片中的电路的复制部分可以根据实现方式而被放置在 1、2 或 3 个芯片中。其它实施例不包括本节的细节或者包括其变体。

[0146] IV. 快照解决方案

[0147] 如上所述，可以使用各种快照技术来获得用于模拟的初始条件信号和输入信号。本快照解决方案不需要被调试逻辑的完整复制，并且通常包括以规则的间隔存储所选设计寄存器的快照。一种方法为设计模块中的储存元件创建具有预定深度的一系列调试储存寄存器（快照寄存器）。原始设计模块中的每个储存元件连接到一组新创建的储存元件。以周期性为基础，设计模块中的储存元件的值被选通到快照寄存器中。快照寄存器中的数据包含被测试设计模块中的寄存器在发生触发条件之前的可确定时间的初始条件。此数据随后被收集，并作为输入提供给软件模拟程序以用于调试。下面描述快照解决方案的某些实现方式。

[0148] 图 21 示出了作为图 1 中的芯片 14 和图 2 中的芯片 38 的示例的芯片 414。参照图 21，芯片 414 包括设计存储器和复制存储器 426、设计寄存器和快照寄存器 428、以及在工作时从逻辑 422-1……逻辑 422-N 接收输入信号的输入 FIFO 430。所述输入信号可以包括各种形式的信号，包括数据、地址、控制以及使能信号。所述输入信号还可包括时钟信号，但该时钟信号在其它图中示出。响应于诸如错误或其它指定条件的触发条件（或事件），触发逻辑 418 提供促使存储器 426、寄存器 428 以及 FIFO 430 中的某些或所有单元冻结其操作的触发信号。如下面详细地解释的那样，存储器 426 和寄存器 428 通过信号 TdoMs 和 TdoRs 来提供初始条件，且 FIFO 430 响应于来自 JTAG 控制器 420 的信号通过例如接口 16 向计算机 20（在图 1 中）提供输入信号以供模拟器使用。一方面，复制存储器包含被延迟存储器输入 FIFO 延迟的存储器数据，意指复制存储器的内容是先前的若干时钟周期处的设计存储器的内容，所述若干时钟周期是存储器输入 FIFO 的长度。另一方面，快照寄存器包含规则间隔的设计寄存器的快照数据，所述规则间隔由快照脉冲来控制。

[0149] 存在各种实现触发信号动作的方式。例如，所述触发信号可以包括多个信号，其中每个信号冻结不同的组件。具有多个信号的优点是可以使其被声明为用于不同的时间长度。

[0150] 图 22 示出了设计存储器和复制存储器 426 中的设计和复制存储单元 434-1……434-N，该设计和复制存储单元 434-1……434-N 接收输入信号 1……输入信号 N 以及时钟、触发和 JTAG 信号。时钟信号对于每个存储单元来说可以是相同的，或者对于某些或全部存储单元来说可以是不同的。同样地，触发和 JTAG 信号对于每个存储单元来说可以是相同的，或者对于某些或全部存储单元来说可以是不同的。存储单元 434-1……434-N 提供输出信号 TdoM1……TdoMN，其中的某些或全部可以用来提供初始条件信号。

[0151] 图 23 示出了设计寄存器和快照寄存器 428 中的设计和快照寄存器单元 436-1……

436-N, 该设计和快照寄存器单元 436-1……436-N 接收输入信号 1……输入信号 N 以及时钟、触发和 JTAG 信号。时钟信号对于每个寄存器单元来说可以是相同的, 或者对于某些或全部寄存器单元来说可以是不同的。同样地, 触发和 JTAG 信号对于每个寄存器单元来说可以是相同的, 或者对于某些或全部寄存器单元可以是不同的。寄存器单元 436-1……436-N 提供输出信号 TdoR1……TdoRN, 其中的某些或全部可以用来提供初始条件信号。请注意, 到存储单元 434-1 中的输入不一定与到寄存器单元 436-1 中的输入匹配。此外, 图 21、图 22 和图 23 中的字母 N 的值不一定是相同的, 并且在实际中通常是不同的。例如, 图 23 中的 N 可以比图 21 和图 22 中的 N 大得多。除所示的那些之外, 可以从许多寄存器提供初始条件信号。

[0152] 图 21、图 22 和图 23 本质上是示意性的, 以传达某些实施例的某些方面, 而不提供完整的电路布局。图 21、图 22 和图 23 所示的那些组件之间可能存在中间电路。例如, 来自逻辑 422-1 的相同信号不一定被直接施加于存储器 426 和寄存器 428。其它实施例包括不同的细节。

[0153] 快照解决方案与存储器和寄存器一同工作, 所述存储器和寄存器接收具有相同频率的时钟信号, 或接收具有不同频率的时钟。下面是在存储器 426、寄存器 428 以及 FIFO 430 中的某些存储单元、寄存器单元以及 FIFO 接收与存储器 426、寄存器 428 以及 FIFO 430 中的其它存储单元、寄存器单元以及 FIFO 不同的时钟信号的情况下, 图 21 中的芯片 414 的示例。在以下示例中, 时钟 A 具有高于时钟 B 的频率。

[0154] 图 24 提供了设计和复制存储单元 446 和 448 的一种示例, 其为图 22 中的单元 434-1……434-N 中的两者的示例。存储单元 446 包括设计存储器 A1、存储器输入 FIFO 450、再调用控制电路 454、复制存储器 A1 以及串行输出控制电路 456。存储单元 448 包括设计存储器 B1、存储器输入 FIFO 460、再调用控制电路 464、复制存储器 B1 以及串行输出控制电路 466。

[0155] 设计存储器 A1 和 FIFO 450 从逻辑 A1 接收存储器输入信号, 逻辑 A1 是图 21 的逻辑 422-1……422-N 中的一者或多者。由时钟 A 来对设计存储器 A1 和 FIFO 450 进行时钟控制。在触发条件之前, 也由时钟 A 来对复制存储器 A1 进行时钟控制。再调用控制电路 454 从 FIFO 450 接收信号, 并将其传递到复制存储器 A1。FIFO 450 的深度可以是最大再调用倒退长度, 虽然在下文的解释中, 倒退的实际周期数可以小于最大倒退长度, 但是所述最大再调用倒退长度是可以倒退的最大周期数。一方面, 最大再调用倒退长度是快照脉冲的间隔, 表示快照寄存器内容的周期。然而, 触发信号可以在两个快照脉冲之间的任何时间发生, 因此, 可以倒退的时钟周期的数目在 1 与最大再调用倒退长度之间变化。为了保证最小数目的倒退周期, 分两阶段来设计快照寄存器 (参见图 25 和以下说明), 且 FIFO 450 的深度是最大再调用倒退长度的两倍。例如, 对于具有两倍最大再调用倒退长度的 FIFO、两阶段快照寄存器 (存储 1000 时钟前的快照内容的阶段 1, 和存储 2000 时钟前的快照内容的阶段 1) 且由时钟 A 的 1000 个周期构成的快照脉冲周期, 信号在时钟 A 的 2000 个周期内穿过 FIFO 450。用此设计, 阶段 1 快照内容存储触发之前的 1 与 1000 时钟周期之间的时钟周期的值。阶段 2 快照内容存储触发之前的 1000 与 2000 时钟周期之间的时钟周期的值。复制存储器内容存储触发之前的 2000 个时钟周期的设计存储器内容。

[0156] 本文所使用的“时钟周期”并非必须等于时钟周期的时间段 (period)。例如, 根

据实现方式,在双倍数据速率系统中,每时钟时间段可以存在两个时钟周期,并且在四倍数据速率系统中,每个时钟时间段可以存在四个时钟周期。然而,这对于每种实现方式是专用的。在某些实现方式中,即使是双倍数据速率或四倍数据速率时钟,时钟周期也可等于时钟信号的时间段。

[0157] 同样地,设计存储器 B1 和 FIFO 460 从逻辑 B1 接收存储器输入信号,逻辑 B1 是图 21 的逻辑 422-1……422-N 中的一者或多者。由时钟 B 来对设计存储器 B1 和 FIFO 460 进行时钟控制,时钟 B 在本示例中具有低于时钟 A 的频率。在触发条件之前,也由时钟 A 来对复制存储器 B1 进行时钟控制。FIFO 460 的深度被选择为与时钟 B 和时钟 A 的频率成比例,以便信号以与信号穿过 FIFO 450 相同的时间量穿过 FIFO 460(虽然根据 FIFO 450 的频率和深度,信号以完全相同的时间量穿过也许是不可能的)。再调用控制电路 464 从 FIFO 460 接收信号,并将其传递到复制存储器 B1。FIFO 450 和 460 可以具有分别等于输入 A1 和 B1 的宽度的宽度。

[0158] 请注意,设计存储器 A1 和 B2 及复制存储器 A1 和 B1 并非必须在时钟 A 和 B 的每个周期将信号写入其中,而是仅仅在满足其它条件(诸如,存在写入存储器的命令)时的那些周期才将信号写入其中。存储器可以具有使能输入(未示出)。因此,存储器输入信号 A1 和 B2 并非必须在每个时钟周期将信号写入存储器中。

[0159] 当发生触发条件时,触发逻辑 418(在图 21 中)提供促使 FIFO 450 和 460、再调用控制电路 454 和 464 以及复制存储器 A1 和 B1 冻结(不再对时钟进行响应)的触发信号。在本示例中,如所讨论的那样,当组件被冻结时,复制存储器 A1 落后于设计存储器 A1 2000 个时钟,且复制存储器 B1 落后于设计存储器 B1 若干时钟。如下文将描述的那样,JTAG 信号促使复制存储器 A1 和 B1 的某些或全部内容通过串行输出控制电路 456 和 466 作为 TdoMemA1 和 TdoMemB1 信号被写入,TdoMemA1 和 TdoMemB1 信号是图 22 中的信号 TdoM1……TdoMN 的示例。

[0160] 图 25 示出了两个示例性设计和快照寄存器单元 476 和 478,其为图 23 中的设计和快照寄存器单元 436-1……436-N 的示例。寄存器单元 476 包括设计寄存器 A1、阶段 1 寄存器 488、阶段 2 寄存器 492 以及串行输出电路 496。寄存器单元 478 包括设计寄存器 B1、阶段 1 寄存器 508、阶段 2 寄存器 512 以及串行输出电路 516。计数器 482、过零点检测电路 484 以及计数器阅读器和串行输出电路 490 为寄存器单元 376 和 478 提供服务。请注意,单个触发信号用于冻结寄存器单元 476 和 488 的某些组件。在图 23 中,可以存在一个触发信号或多于一个的触发信号。阶段 1 和阶段 2 寄存器可以被称为快照寄存器。寄存器单元 476 接收时钟 A 且设计寄存器 B1 接收时钟 B,而初始条件输出 TdoRegA1 和 TdoRegB1 可以被同步。设计寄存器 A1 和 B1 将其寄存器值提供给逻辑,该逻辑可以与其它所示逻辑相同或不同。在某些实施例中,信号在时钟 A 和 B 的每个周期被写入设计寄存器 A1 和 B1 中(虽然可以重复相同的值),而在其它实施例中,只有当满足诸如使能信号的条件时,信号才被写入设计寄存器 A1 和 B1 中。

[0161] 响应于时钟 A 的时钟周期,设计寄存器 A1 将来自诸如图 21 中的逻辑 422-1……422-N 之一的逻辑提供的信号 A1 的比特位(bit)锁存。响应于时钟 B 的时钟周期,设计寄存器 B1 将来自诸如图 21 中的逻辑 422-1……422-N 之一的逻辑提供的信号 B1 的比特位锁存。设计寄存器 A1 的比特位可用于逻辑和阶段 1 寄存器 488。阶段 1 寄存器 488 中的

比特位可用于阶段 2 寄存器 492。阶段 2 寄存器 492 的比特位可用于响应于 JTAG 信号而工作的串行输出电路 496。设计寄存器 B1 的比特位可用于逻辑（其可以不同于接收设计寄存器 A1 的比特位的逻辑）和阶段 1 寄存器 508。阶段 1 寄存器 508 的比特位可用于阶段 2 寄存器 512。阶段 2 寄存器 512 的比特位可用于响应于 JTAG 信号而工作的串行输出电路 516。

[0162] 虽然设计寄存器 A1 和 B1 可以分别在时钟 A 和时钟 B 的每个周期从寄存器输入信号 A1 和 B1 接收比特位,但阶段 1 和 2 寄存器 488、492、508 和 512 在它们通过来自过零点检测电路 484 的寄存器使能信号而被使能时,只锁存新信号。此使能如下发生。计数器 482 在时钟 A 的每个周期进行计数,并具有等于再调用长度的计数长度。在本示例中,再调用长度是计数长度 1000 (在本示例中,即为时钟 A 的 1000 个时钟周期)。当计数器 482 从计数长度倒数至零时,过零点检测电路 484 声明再调用寄存器使能信号,该信号使得阶段 1 寄存器 488 和 508 及阶段 2 寄存器 492 和 512 被使能。计数器 482 在达到零之后,回到全深计数 (full-depth count) 减一。因此,时钟 A 的每 1000 个时钟周期,阶段 1 和 2 寄存器 488 和 492 接收前一阶段的寄存器值。也就是说,阶段 2 寄存器 492 接收阶段 1 寄存器 488 的寄存器值,且阶段 1 寄存器 488 接收设计寄存器 A1 的寄存器值。因此,在过零点检测电路 484 提供使能信号时,阶段 1 寄存器 488 将保持设计寄存器 A1 的当前值,且阶段 2 寄存器 492 将保持先前的 1000 个时钟周期的设计寄存器 A1 的寄存器值。同样地,当声明使能信号时,阶段 1 寄存器 508 将设计寄存器 B1 的当前寄存器值锁存,且阶段 2 寄存器 512 接收阶段 1 寄存器 508 的寄存器值,该寄存器值为先前的 1000 个时钟 A 周期的寄存器 B1 的寄存器值。

[0163] 当发生触发条件时,触发信号冻结计数器 482,以使得阶段 1 和 2 寄存器 488、492、508 和 512 的寄存器值不变。作为一种示例,假设过零点检测电路 482 启用寄存器使能信号,以使得阶段 1 和 2 寄存器 488、492、508 和 512 被更新。然后,触发信号之后的 55 个时钟冻结计数器 482,因此计数器的计数为 $1000-1-55 = 944$ 。根据计数 55,系统可以知道阶段 1 寄存器 488 和 508 中的值来自于先前的 55 个时钟周期,且阶段 2 寄存器 492 和 512 中的值来自于先前的 1055 个时钟周期。计数器读取和串行输出电路 490 提供该计数值以供控制电路使用 (图 25)。

[0164] 图 25 中的触发信号与图 24 中相同,因此存储器 A1 和 B1 在与阶段 1 和 2 寄存器 488、492、508 和 512 相同的时间被冻结。由于存储器输入 FIFO 是指定数目的时钟周期的深度的两倍,所以复制存储器的内容与从触发时间之前的 2000 个周期起的设计存储器中的内容相同。在某些实施例中,时钟周期的数目可以由设计者指定。由于阶段 2 寄存器 492 具有从先前的 1055 个周期起的设计寄存器 A1 的寄存器值,所以 JTAG 控制信号通过使 FIFO 450 前进 945 个时钟周期 ($2000-1055 = 945$) 来使复制存储器 A1 的内容前进相同的时钟周期,以便将这 945 个时钟周期内的 FIFO 450 的内容通过再调用控制电路 454 提供给复制存储器 A1。请注意,不是全部 (如果有的话) 这些存储器时钟周期都真正执行存储器写周期——这意味着存储器不一定在每个时钟周期都发生改变。在 945 个周期结束时,复制存储器 A1 是从先前的 1055 个周期起的设计存储器 A1 的快照。这与阶段 2 寄存器 492 和 512 的时钟周期时间相匹配。

[0165] 图 25 所示的电路示出了一种示例性实施例,且其它变体在本发明的范围内。例

如,可以由时钟 B 来对快照寄存器进行时钟控制,或者可以由没有使能端口的快照脉冲发生器来对快照寄存器进行时钟控制。

[0166] 图 26 示出了由时钟 A 进行时钟控制的时钟 A 域输入 FIFO 520 和由时钟 B 进行时钟控制的时钟 B 域输入 FIFO 522。FIFO 520 接收被图 24 中的设计存储器 A1 和 FIFO 450 及图 26 中的设计寄存器 A1 接收到的存储器数据。FIFO 522 接收被图 24 中的设计存储器 B1 和 FIFO 460 及图 26 中的设计寄存器 B1 接收到的存储器数据。FIFO 520 的深度可以与图 24 中的 FIFO 450 的深度相匹配,且 FIFO 522 的深度可以与图 26 中的 FIFO 460 的深度相匹配。在本示例中,FIFO 520 和 522 的输出不对设计逻辑进行馈送,但在其它实施例中可以。

[0167] FIFO 520 和 522 响应于触发信号而停止接收输入信号。来自 JTAG 控制逻辑 420(图 21)的 JTAG 控制信号促使某些输入数据通过串行输出控制 524 而作为 TdoA 输入和 TdoB 输入被写出。触发条件之后的时间内的 FIFO 520 和 522 的内容被提供作为输入信号。在本示例中,这将是最近 1055 个周期的 FIFO 520 的输入信号和相关周期数目的 FIFO 522 的输入信号。在某些实施例中,这可以包括指针或其它机制,以使得 FIFO 520 前进 945 个周期以及 FIFO 522 前进一定量到达同一位置。

[0168] 这里,可以在 JTAG 控制下提供复制存储器 A1 和 B1 及阶段 2 寄存器 492 和 512 的内容作为初始条件信号,且可以在 JTAG 控制下提供输入 FIFO 520 和 522 的内容作为输入信号。可以存在提供初始条件信号的附加部分的附加复制存储器(或其它形式的存储器)和阶段 2 寄存器,并且可以存在附加 FIFO 以提供用于特定模拟的附加输入信号。请注意,还可以以与对上文讨论的存储器所做的相同的方式来复制用户设计中的诸如 Xilinx SRL(移位寄存器)的其它存储器。在某些实施例中,可以控制通过由 JTAG 控制信号来操作 SRL 的输入而获取 SRL 的内容。例如,SRL 的内容也可以落后于设计 SRL 2000 个时钟。

[0169] 即使由时钟 A 来对图 24-图 26 中的某些所示存储器、寄存器单元及 FIFO 进行时钟控制,且由时钟 B 来对其它各组件进行时钟控制,也不存在使不同组件同步的问题。相反,仅仅输出初始条件和输入信号以供使用。

[0170] 存在可以实现快照解决方案的各种方式。例如,包括与图 26 的电路类似的电路的系统可以只包括单阶段的快照寄存器(亦即,只有阶段 1 寄存器 488,而没有阶段 2 寄存器 492)。包括与图 26 的电路类似的电路的系统可以包括多于两阶段的快照寄存器(例如阶段 1 寄存器 488、阶段 2 寄存器 492 以及阶段 3 寄存器(未示出))。在以上示例中,FIFO 450 是最大再调用倒退长度的两倍,但在其它实施例中,其可以是诸如大于最大再调用倒退长度的两倍的其它长度。

[0171] 快照解决方案的一种示例性实施例包括用于设计模块的所选寄存器(或存储器或任何储存元件)的快照寄存器。一方面,复制所有寄存器。另一方面,只复制所选寄存器。复制模块还可包括例如 FIFO 的延迟电路,以存储设计模块的输入数据。延迟电路的深度应适用于快照脉冲的周期。如果对于每个设计寄存器存在多个快照寄存器,则延时电路的深度是多个快照周期。还可以提供用于设计模块的所选存储器部分的复制电路、用于访问存储器内容的控制电路以及用于复制寄存器的输入的可选延迟电路。当发生触发条件时,存储器和快照寄存器可能不相互关联。在这种情况下,对存储器内容进行时钟控制若干时钟周期以便与快照寄存器的时序相匹配。

[0172] 图 27 示出了可以在某些快照解决方案中使用的某些可选电路。参照图 27, 考虑包括诸如寄存器或存储元件的原始设计, 例如触发器后面跟着逻辑和另一触发器, 类似于触发器 544、逻辑 548 以及触发器 552。提供图 27 的附加组件以实现快照解决方案。

[0173] 输入信号被提供给触发器 544 和 FIFO 542, 其中两者都由时钟信号来进行时钟控制的。当然, 可以存在许多信号通路, 其中触发器 544 只处理其中之一。触发器 544 的输出被提供给逻辑 548, 逻辑 548 的输出被提供给触发器 552, 触发器 552 提供数据输出信号。触发器 552 和 544 的输出被提供给触发器 554 和 556, 触发器 554 和 556 又将其提供给触发器 558 和 560。快照脉冲对触发器 554、558、556 和 560 进行时钟控制。在某些实施例中, 设计者可以选择快照脉冲的频率。一旦检测到触发条件, 则快照脉冲停止。响应于触发信号, 触发器 558 和 560 的输出提供初始条件信号, 且 FIFO 542 保持用于模拟器的输入信号。初始条件信号和输入信号可以被扫描输出到计算机或其它外部测试设备, 并经重新格式化且呈现给模拟器以用于电路模拟。

[0174] 快照解决方案存在的一个问题是触发条件可能在任何时间发生。同时, 快照脉冲可能但不一定以规则的间隔发生。如果缓冲器深度是 256, 则快照脉冲将每 256 个时钟周期发生一次。触发可能在快照脉冲之后的第 1 与第 255 个时钟周期之间的任何时间发生。如果只存在一个触发器缓冲器, 则可能存在多达 255 个时钟周期, 但通常在触发条件之前将较少。通过加入触发器 558 和 560, 可管线传输 (pipeline) 另一快照周期 (在这种情况下为 256 个周期) 的快照结果, 以便收集 256 与 512 周期之间的样本。FIFO 542 的深度同样可以是双倍的。还可以提供附加电路以快速转发存储器内容, 从而与快照寄存器的时序相匹配。

[0175] 可以在具有不同于上文所示和所述的某些细节的各种其它实施例中实现快照解决方案。

[0176] 快照解决方案可以与第 III 节中所述的特征相结合地使用。某些节的特征还可以在同一芯片中使用, 但是它们可能不直接是快照方法的一部分。然而, 快照解决方案的某些实施例的附加特征可以包括通过将 RAM 的输出路由到 FIFO 中来处理设计 (原始) 逻辑中的各种形式的 RAM 的方法。在某些实施例中, FIFO 可以被系统计算机读取, 并在软件模拟中用作来自 RAM 的输出。

[0177] V. 时分复用器解决方案

[0178] 存在各种用以实现时分复用器 (TDM) 解决方案的方式。本方法创建时分复用信道系统, 并使用第二信道来保持延迟复制逻辑的所有寄存器和存储元件的状态。在原始设计模块中可以包括用于 TDM 解决方案的附加逻辑, 且电路被复用器 / 解复用器对围绕 (surrounded)。来自原始通路的数据被连接到复用器的一个输入端, 而相同但通过延迟 FIFO 而被延迟的数据通路被连接到另一输入端。复用器将原始数据和延迟数据两者组合成同一通过设计逻辑的数据通路, 所述设计逻辑通过时分复用来运行。通常, 使用以两倍速度运行的时钟来对用于所述数据和延迟数据两者的设计逻辑进行时钟控制。例如, 在每个偶数周期, 原始数据被复用到设计模块管线中, 而在每个奇数周期, 延迟数据被复用输入。解复用器将数据分离回到其各自的数据流中。当检测到触发条件时, 电路被冻结且延迟输出数据流处的值被捕捉而充当用于调试模拟的初始条件。

[0179] 在一种实施例中, 在时分复用器解决方案中, 可以提供如下与触发条件相关的初

始条件和输入信号。诸如通过 FIFO 来延迟输入信号。将输入信号和延迟输入信号复用,并提供给逻辑且随后解复用。延迟输出信号包括初始条件信号。可以在模拟器中检验输入信号以及来自延迟输出信号的初始条件信号,以便例如识别触发条件的原因。时分复用器解决方案可以被称为时间克隆方案。不需要复制逻辑,其结果是节省相当多的芯片空间和设计时间。

[0180] 作为一种示例,图 28 示出了作为图 1 中的芯片 14 和图 2 中的芯片 38 的示例的芯片 614。图 29 提供了用于图 28 的某些实施例中的某些信号的时序图,且图 30 提供了用于图 28 的其它实施例中的某些信号的时序图。

[0181] 参照图 28 和图 29,设计电路(或原始电路)包括寄存器 624、逻辑 638 和寄存器 644。为了提供初始条件和输入信号,由修订设计电路并对设计电路进行添加的至少一个程序(例如如图 3,方框 52)来添加 FIFO 628、复用器电路 634、解复用器电路 642、寄存器 648 和互连线路(interconnection)。该程序例如通过 JTAG 来提供对 FIFO 628 和寄存器 648 的内容的外部访问以供模拟程序使用。

[0182] 寄存器 624 和 FIFO 628 接收数据输入信号(数据输入)。FIFO 628 可以具有等于输入信号的宽度的宽度和对输入信号提供期望延迟的深度。寄存器可以是触发器。寄存器 624 提供单元(cell)(或可能的跳变之间的时间)T1i、T2i、T3i 等中的信号,其中 T1i 意指将被输入到逻辑 638 中的第一数据单元。FIFO 628 提供单元 DT1i、DT2i、DT3i 等中的信号,其中 DT1i 意指将被输入到逻辑 638 中的第一延迟数据单元。可以看出,单元 T1i、T2i、T3i 等和 DT1i、DT2i、DT3i 等的长度均是两个时钟周期。如果寄存器 624 和 FIFO 624 每隔时钟 1 的一个周期进行响应,则会发生这种情况。可以添加电路以促使这种情况发生。

[0183] 由于 FIFO 628 使信号延迟,因此从 FIFO 628 中输出的初始延迟长度内的信号是无效数据。然而,如果直至初始延迟完成之前,未发生触发条件,则这将是问题。为了便于说明,在图 29 中,通过 FIFO 628 实现的延迟仅仅是时钟 1 的两个时钟周期。然而,在实际中,延迟可能比两个时钟周期大得多。例如,其可以是 1000 个时钟周期或者更高或更低数目的时钟周期。

[0184] 复用器 634 在例如奇数时钟周期期间将来自寄存器 624 的信号传递到逻辑 638,并在偶数时钟周期期间将来自 FIFO 628 的信号传递到逻辑 638。可以看出,从复用器 634 出来的信号具有是数据单元之间的跳变频率,该频率是从寄存器 624 和 FIFO 628 输出的信号的两倍。解复用器 642 在例如奇数时钟周期期间将来自逻辑 638 的信号(T1out、T2out 等)传递到寄存器 644,并在偶数时钟周期期间将来自逻辑 638 的信号(无效、DT1out、DT2out 等)传递到寄存器 648。通常,不能同时对寄存器 644 和 648 进行时钟控制,因此需要使一个停留(clock off)在时钟负沿。从图 29 中可以看出,从解复用器 642 出来的信号具有的单元长度是进入解复用器 642 中的信号的两倍。根据实现方式,可以出于各种目的来提供寄存器 644 的输出。响应于触发信号,FIFO 628 和寄存器 648 冻结。通过例如 JTAG 接口,寄存器 648 的内容可以被用作作用于模拟器的初始条件信号 Tdos1,且 FIFO 628 的内容可以被用作作用于模拟器的输入信号 Tdos2。根据实现方式,将数据从复用器 634 通过逻辑 638 传递到解复用器 642 可能只花费一个时钟周期。在其它实施例中,可能花费多于一个时钟周期。在某些实施例中,在触发信号被施加于寄存器 648 之前可能存在一个或两个周期延迟。可以在判定哪些信号是初始条件信号时考虑该延迟。

[0185] 并不是所有实施例都需要图 28 和图 29 的细节。例如,在某些实施例中,通常包含任何寄存器或状态信息的逻辑 638 能够处理相当于 2X 时钟的数据速率。在某些实施例中,也由 2X 时钟来对复用器 634 和 / 或解复用器 642 进行时钟控制。根据实现方式,其它组件也可以接收更高频率的时钟。可以使用双倍或四倍数据速率的信令。

[0186] 图 30 提供了其中将 2X 时钟 1 信号用于进行时钟控制的图 28 的各种实现方式之一的时序图。在图 30 中, T1i、T2i 等及无效、DT1i、DT2i 等信号单元的长度等于时钟 1 的周期。复用器 634 提供信号 T1i、无效、T2i、无效、T3i、DT1i、T4i、DT2i 等,其中,这些单元具有等于 2X 时钟 1 的一个周期的长度。T1out、T2out 等及无效、无效、DT1out、DT2out 信号具有长度等于时钟 1 的周期的单元。在某些实施例中,图 29 和图 30 的信号在技术上不包括单元,但附图仍然示意性地示出了位于可能的跳变之间的时间。

[0187] 图 31 示出了由时钟 1 进行时钟控制的时分复用器电路 650 和由时钟 2 进行时钟控制的另一时分复用器电路 652。时钟 1 和 2 可以是异步的,且具有不同的频率。可选地,时钟 1 和 2 可以是相同的,或具有不同的频率,其中一个频率是另一个频率的整数倍。电路 650 与图 28 中的电路 620 相同,但是添加了对下降沿进行响应的寄存器 630。寄存器 630 是可选的,其仅仅示出另一种可选方式。电路 652 包括寄存器 654、接收输入信号 2 (数据输入) 的 FIFO 658 以及 (在某些实施例中) 寄存器 660。复用器 664 将来自寄存器 654 和来自寄存器 660 的信号复用,并将其提供给逻辑 668。解复用器 672 接收来自逻辑 668 的输出信号,并将其解复用以便向寄存器 674 和 678 提供信号。当声明触发信号响应于触发条件时, FIFO 658 和寄存器 678 被冻结,并提供用于模拟程序的输入数据和初始条件。如结合图 28 所述,可以由具有不同频率的时钟来对时分复用器电路 620 的不同组件进行时钟控制。同样地,可以由具有不同频率的时钟对时分复用器电路 650 和 652 进行时钟控制。

[0188] 图 32 示出了时分复用器电路的另一实施例。触发器寄存器 714、逻辑 720 和寄存器 730 为设计电路的一部分。提供图 32 的附加电路是为了收集与触发条件相关的初始条件和输入信号。在某些实施例中,由程序来自动设计图 32 (以及其它图) 的附加电路。在不同的实现方式中,在此过程中存在不同的电路设计者参与。

[0189] 图 33 是图 32 的电路的时序图。参照图 32 和图 33,输入信号 (数据输入) T1、T2、T3、T4 等被复用器 710 和 FIFO 704 所接收,该 FIFO 704 通过信号 TD1、TD2、TD3、TD4 等将延迟数据提供给复用器 710。FIFO 704 具有通过信号接收多重并行数据的宽度。由 2X 时钟信号对触发器寄存器 712 和 714 进行时钟控制。由二次分频电路 706 导出 1X 时钟信号 (可选地,可以通过将 1X 时钟乘以 2 来生成 2X 时钟)。1X 时钟信号对 FIFO 704、复用器 710 和解复用器 724 进行时钟控制。在一种优选实施例中,可以由 2X 时钟对逻辑 720 进行时钟控制,但在其它实施例中,由 1X 时钟对其进行时钟控制。可以由 1X 时钟 (或者在其它实现方式中由 2X 时钟) 对寄存器 730 和 732 进行时钟控制。作为示例,在每个奇数周期中,数据被复用输入,而在每个偶数周期中,延迟数据被复用输入。复用器 710 的输出 (T1、TD1、T2、TD2、T3、TD3) 被提供给寄存器 712 (Q1) 以使输出延迟 2X 时钟的一个周期,并将其提供给寄存器 714,寄存器 714 同样地使其延迟 2X 时钟的另一周期。在任一时刻,管线化 (pipelined) 寄存器 712 和 714 包含来自输入至复用器 710 的数据输入和延迟数据输入这两者的数据。

[0190] 逻辑 720 可以提供相同频率、但被进一步延迟的输出 (T1d、TD1d、T2d、TD2d 等)。

解复用器 724 可以提供数据输出信号 (T1d、T2d、T3d 等), 以及半频率处的延迟数据输出信号 (TD1d、TD2d、TD3d 等), 其分别被提供给寄存器 730 和寄存器 732。图 32 的电路被允许运行至检测到触发条件为止。那时, FIFO 704 和寄存器 732 被冻结并提供用于模拟 / 调试的输入信号和初始条件。可以以各种方式来修改图 32 的时分复用器电路。

[0191] 用时分复用, 可以使用相同的设计逻辑 (例如逻辑 638 或逻辑 720) 来确定实时寄存器和延迟寄存器两者的下一个状态。使用 TDM 方法的设计内的公共元件处理的数据是原始设计内的公共元件处理的数据的两倍。如果通过公共元件的传播延迟显著, 则可以降低时钟速度, 以适当地确保电路功能。可以与在第 III 节中所述的特征相结合地使用 TDM 方案。

[0192] VI. 附加信息和实施例

[0193] 本发明不局限于任何特定类型的信令。输入和时钟信号可以是单端或差动信号。时钟可以包括“监视器 (eye)”。时钟控制可以是单数据速率、双倍数据速率、四倍数据速率等。在双倍数据速率中, 可以使用单时钟信号的上升或下降沿, 或者可以使用两个异相时钟。信号可以被分组或不分组。时钟信号可以在芯片内部或外部生成。其可以与数据信号分开或者被嵌入数据中并从该数据恢复出来。

[0194] 本文所述的用于设计、配置、制造、运行、调制、收集用于模拟的初始条件信号和输入信号的程序 (诸如图 3 中的程序以及结合图 4 和图 5 使用的那些程序) 可以通过因特网 (诸如通过下载信号) 或通过诸如 CD ROM、DVD、闪存或其它存储器的存储介质来提供。一旦被接收到, 程序可以被存储在计算机存储器或其它存储器中。在某些实施例中, 执行芯片中的功能的指令可以被硬线连接到电路。在其它实施例中, 可以通过固件和 / 或软件来发起至少某些功能。此类固件或软件可以通过因特网或通过诸如 CD ROM、DVD、闪存或其它存储器的存储介质来提供。在某些情况下, 逻辑只包括电路, 在其它情况下, 逻辑包括响应于软件或固件而工作的电路。

[0195] 附图本质上是示意性的, 而并非意为表示实际电路布局。此外, 在实际实现方式中, 在芯片中将存在各种附加电路, 并且在附图所示的电路之间可能存在电路。所示的组件可以具有各种附加输入端和输出端。本发明的各种实施例可以包括不同于本文所示和所述的那些细节的细节。

[0196] 虽然寄存器是一种存储器, 但其每次保持一比特位, 而本公开中所提及的存储器每次保持多于一个比特位。在某些图中, 单个寄存器 (或触发器) 被示为接收输入信号。然而, 这些寄存器可以表示接收并行数据比特位的多个并行寄存器。例如, 在图 25 中, 设计寄存器 A1、阶段 1 寄存器 488 以及阶段 2 寄存器 492 均可以表示接收并行信号的多个并行寄存器。

[0197] 在某些实施例中, 寄存器中的比特位可以是多级比特位, 该多级比特位保持多于一比特位的信息。也就是说, 在某种情况下, 信号仅仅具有高压或低压, 而在其它情况下, 信号具有多个水平, 以使每比特位表示多于两个值。上述解决方案可以被修改, 以整合多级信令。

[0198] 在某些实现方式中, FIFO 具有固定深度, 且在其它实现方式中, FIFO 具有在制造或配置之后可以改变的深度。计数器可以倒数至零或另一值, 或者它们可以从零或另一值开始正数。本发明的许多方法可以用数字处理系统来执行, 诸如传统的通用计算机系统。还

可以使用被设计或编程为只执行有限功能的专用计算机。

[0199] 如果本说明书陈述组件、特征、结构或特性“可以”、“可能”或“也许”被包括,则不要求包括该特定组件、特征、结构或特性。如果本说明书或权利要求提及“一个”结构,则其并不意味着只存在一个该结构。

[0200] 本发明的一个实施例可以是电路设计和被实现为被存储在机器可读介质中的计算机程序的合成计算机辅助设计软件,所述机器可读介质诸如为 CD ROM、磁性硬盘或光盘或各种其它可选储存设备。此外,本发明功能的许多方法可以用数字处理系统来实现,诸如传统的通用计算机系统。还可以使用被设计或编程为只执行一种功能的专用计算机。

[0201] 图 34 示出了可以用于本发明的典型计算机系统的一个示例。该计算机系统可以用来执行在 HDL 代码中描述的设计的逻辑合成。请注意,虽然图 34 示出了计算机系统的各种组件,但其并不意图表示将组件互连的任何特定架构或方式,因为此类细节与本发明没有密切关系。应注意的是,图 34 的架构仅仅是出于说明的目的而提供的,并且结合本发明所使用的计算机系统或其它数字处理系统不限于此特定架构。还应认识到,具有较少组件或可能更多组件的网络计算机及其它数据处理系统也可用于本发明。图 34 的计算机系统可以是例如 IBM 或苹果麦金塔 (Macintosh) 计算机。

[0202] 如图 34 所示,作为一种数据处理系统的计算机系统 1101 包括被耦合到微处理器 1103、ROM 1107、易失性 RAM 1105 和非易失性存储器 1106 的总线 1102。可以是来自英特尔、摩托罗拉公司或 IBM 的微处理器的微处理器 1103 被耦合到高速缓冲存储器 1104。总线 1102 将这些各种组件互连在一起,并且将这些组件 1103、1107、1105 和 1106 互连到显示控制器和显示设备 1108 及诸如输入 / 输出 (I/O) 设备的外围设备,所述输入 / 输出 (I/O) 设备可以是鼠标、键盘、调制解调器、网络接口、打印机、扫描仪、视频摄像机及本领域中众所周知的其它设备。通常,输入 / 输出设备 1110 通过输入 / 输出控制器 1109 耦合到系统。易失性 RAM 1105 通常被实现为需要连续供电以刷新或保持存储器中的数据的数据的动态 RAM (DRAM)。非易失性存储器 1106 通常是磁性硬盘驱动器、磁性光驱、光驱、DVD RAM 或即使系统掉电之后也保持数据的其它类型的存储器系统。通常,非易失性存储器还可以是随机存取存储器,虽然对此并无要求。虽然图 34 显示了非易失性存储器是直接耦合到数据处理系统中的其它组件的本地设备,但应认识到,本发明可以利用远离系统的非易失性存储器,诸如通过诸如调制解调器或以太网接口的网络接口耦合到数据处理系统的网络存储设备。如本领域中众所周知的那样,总线 1102 可以包括通过各种桥接器、控制器和 / 或适配器相互连接的一个或多个总线。在一个实施例中,I/O 控制器 1109 包括用于控制 USB 外围设备的 USB (通用串行总线) 适配器和 / 或用于控制 IEEE-1394 外围设备的 IEEE-1394 总线适配器。

[0203] 通过本说明书显而易见的是,本发明的方法可以至少部分地在软件中体现。也就是说,可以在计算机系统或其它数据处理系统中响应于其处理器执行包含在存储器中的指令序列来实现所述技术,所述处理器诸如为微处理器,所述存储器诸如为 ROM 1107、易失性 RAM 1105、非易失性存储器 1106、高速缓冲存储器 1104 或远程存储设备。在各种实施例中,可以与软件指令相结合地使用硬线连接电路以实现本发明。因此,所述技术不限于硬件电路和软件的任何特定组合,也不限于由数据处理系统执行的指令的任何特定源。另外,通过本说明书,各种功能和操作被描述为由软件代码来执行或引起,以简化说明。然而,本领域

技术人员将认识到此类表达的意思是所述功能是由诸如微处理器 1103 的处理器执行代码而引起的。

[0204] 可以使用机器可读介质来存储软件和数据,所述软件和数据在被数据处理系统执行时促使系统执行本发明的各种方法。此可执行软件和数据可以被存储在包括例如 ROM 1107、易失性 RAM 1105、非易失性存储器 1106 和 / 或高速缓冲存储器 1104 在内的各种不同位置中。此软件和 / 或数据的一部分可以被存储在这些存储设备中的任何一者中。

[0205] 因此,机器可读介质包括以机器(例如,计算机、网络设备、个人数字助理、制造工具、具有一组的一个或多个处理器的任何设备等等)可访问的形式提供(即,存储和 / 或发送)信息的任何机构。例如,机器可读介质包括可记录 / 非可记录媒介(例如,只读存储器 (ROM)、随机存取存储器 (RAM)、磁盘储存介质、光学储存介质、闪存设备等等)以及信号传播的电学、光学、声学或其它形式(例如,载波、红外信号、数字信号等)等。

[0206] 虽然已根据多个实施例描述了本发明,但本发明不应只限于所述的那些实施例,而是可以在不脱离随附权利要求的精神和范围的修改和变更的情况下来实施。因此,应将本说明书视为说明性而不是限制性的。

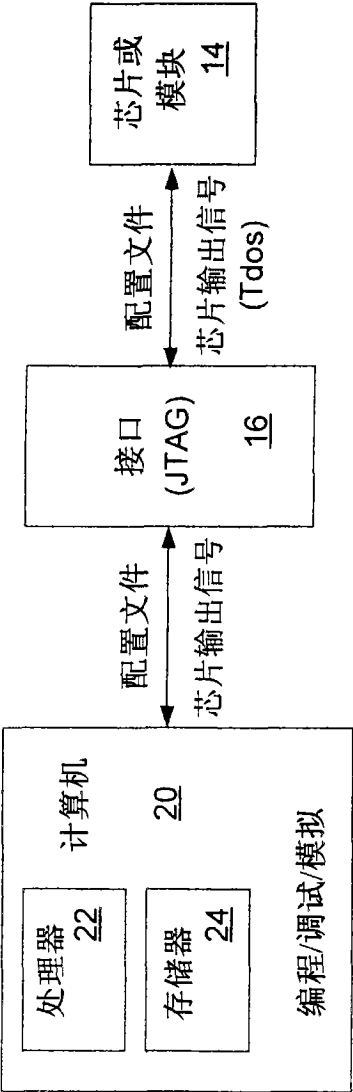


图 1

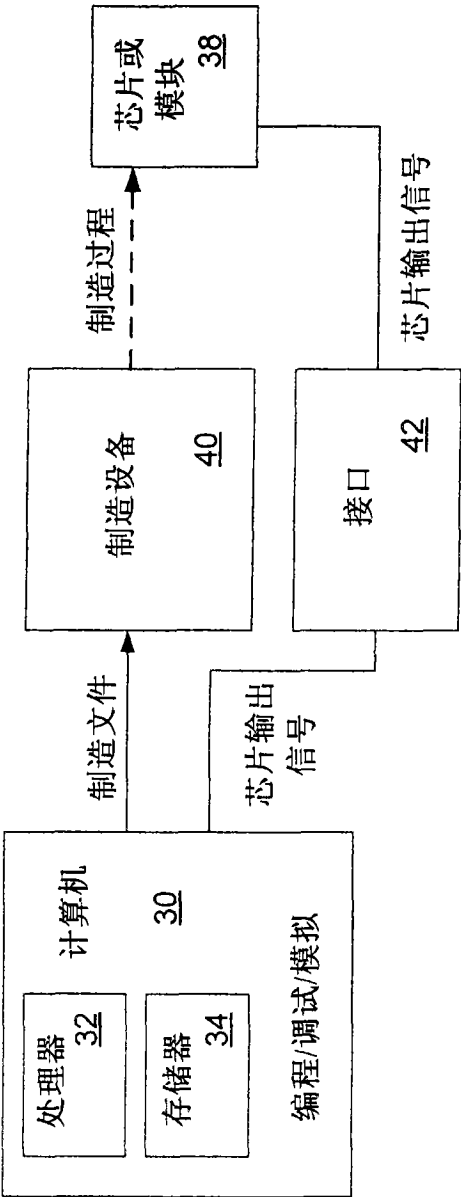


图 2

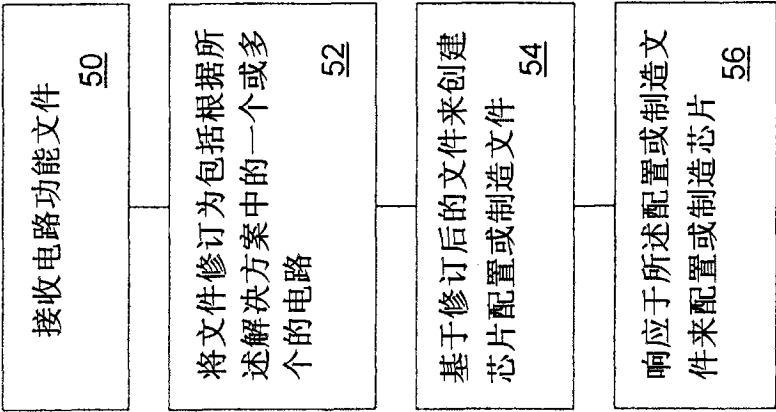


图 3

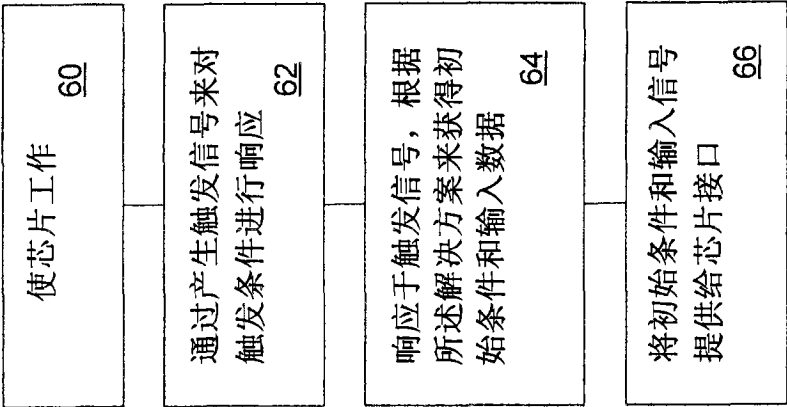


图 4

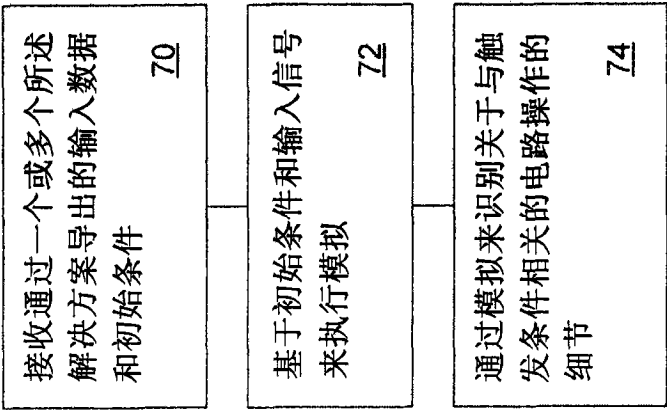


图 5

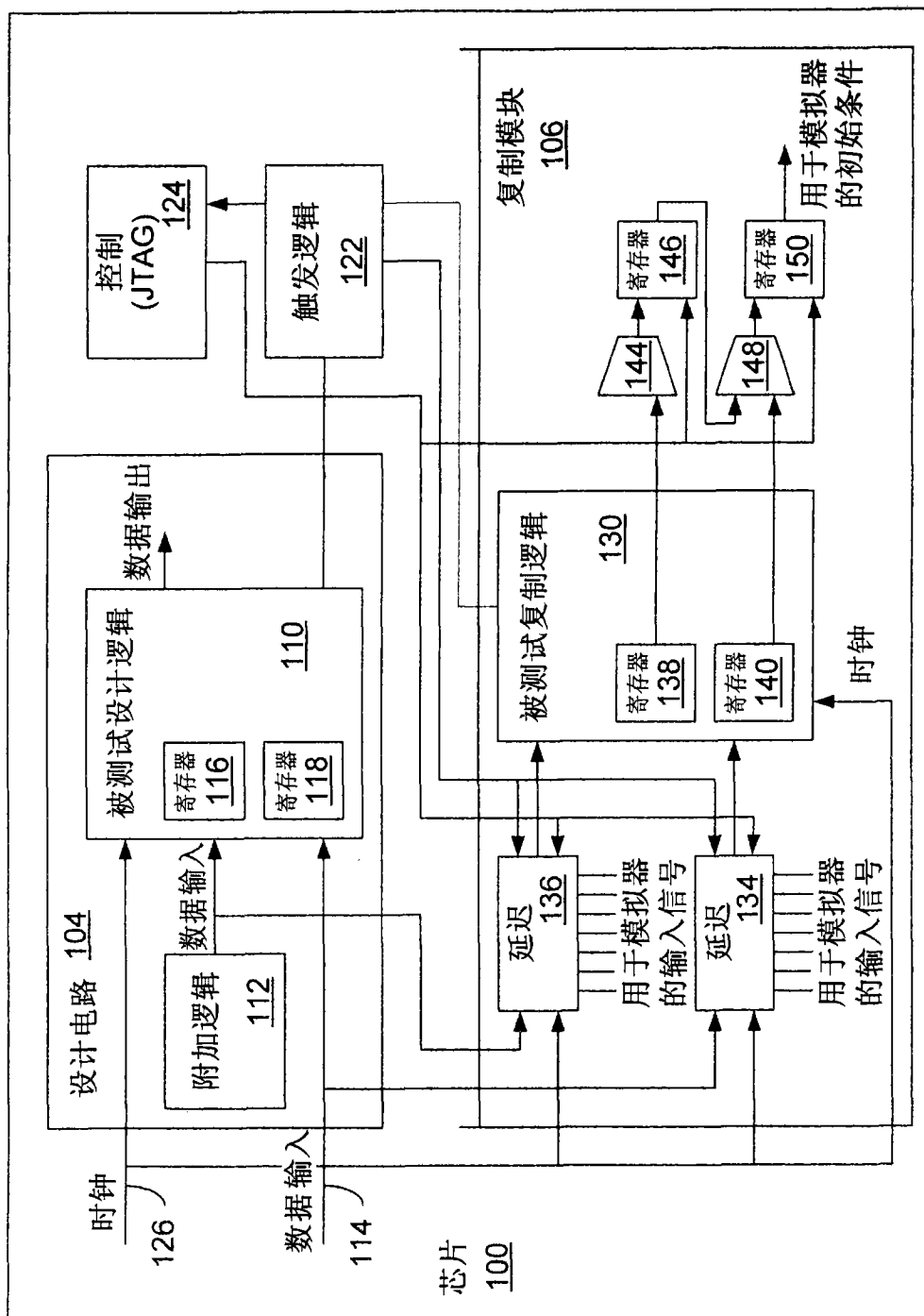


图 6

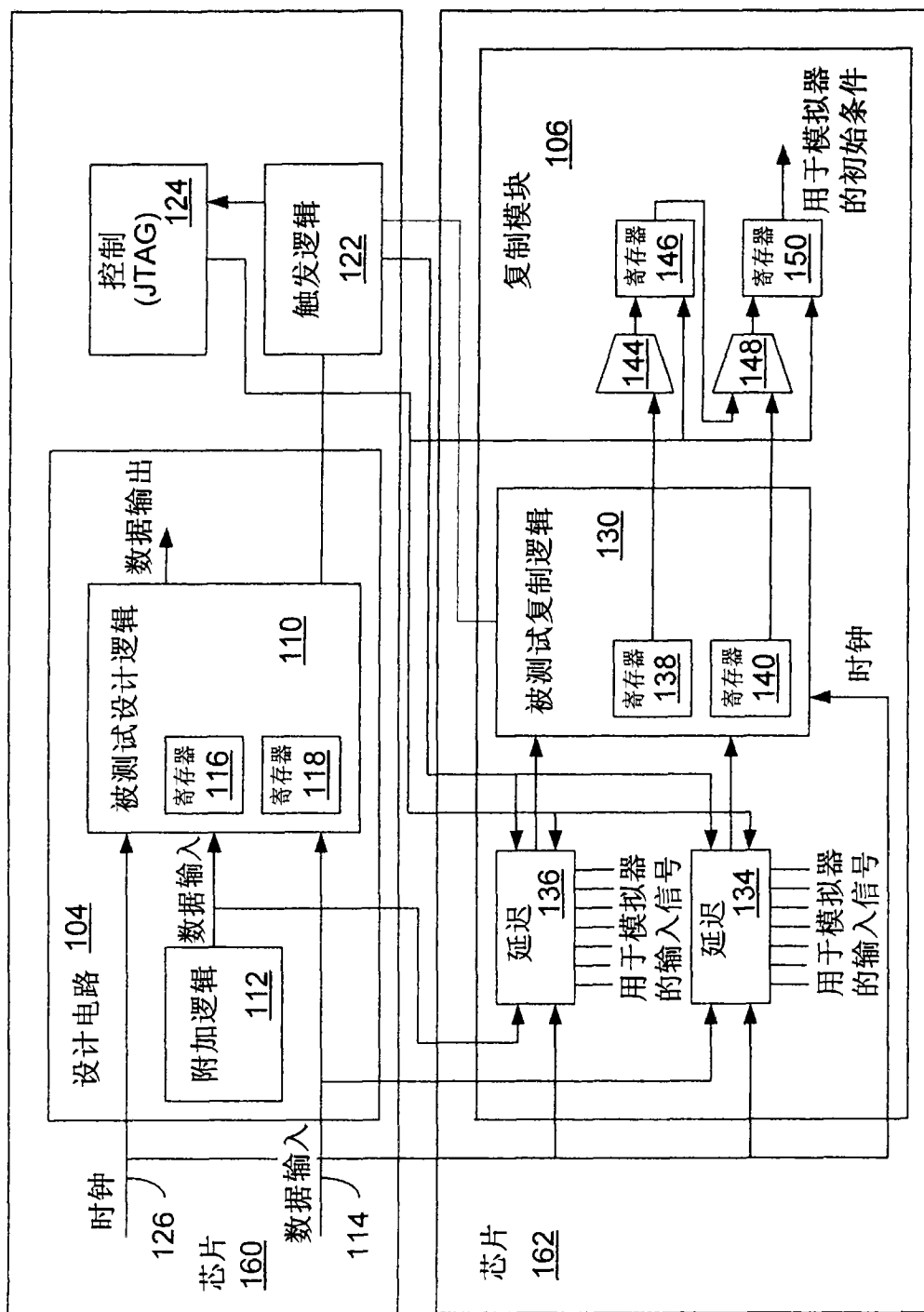


图 7

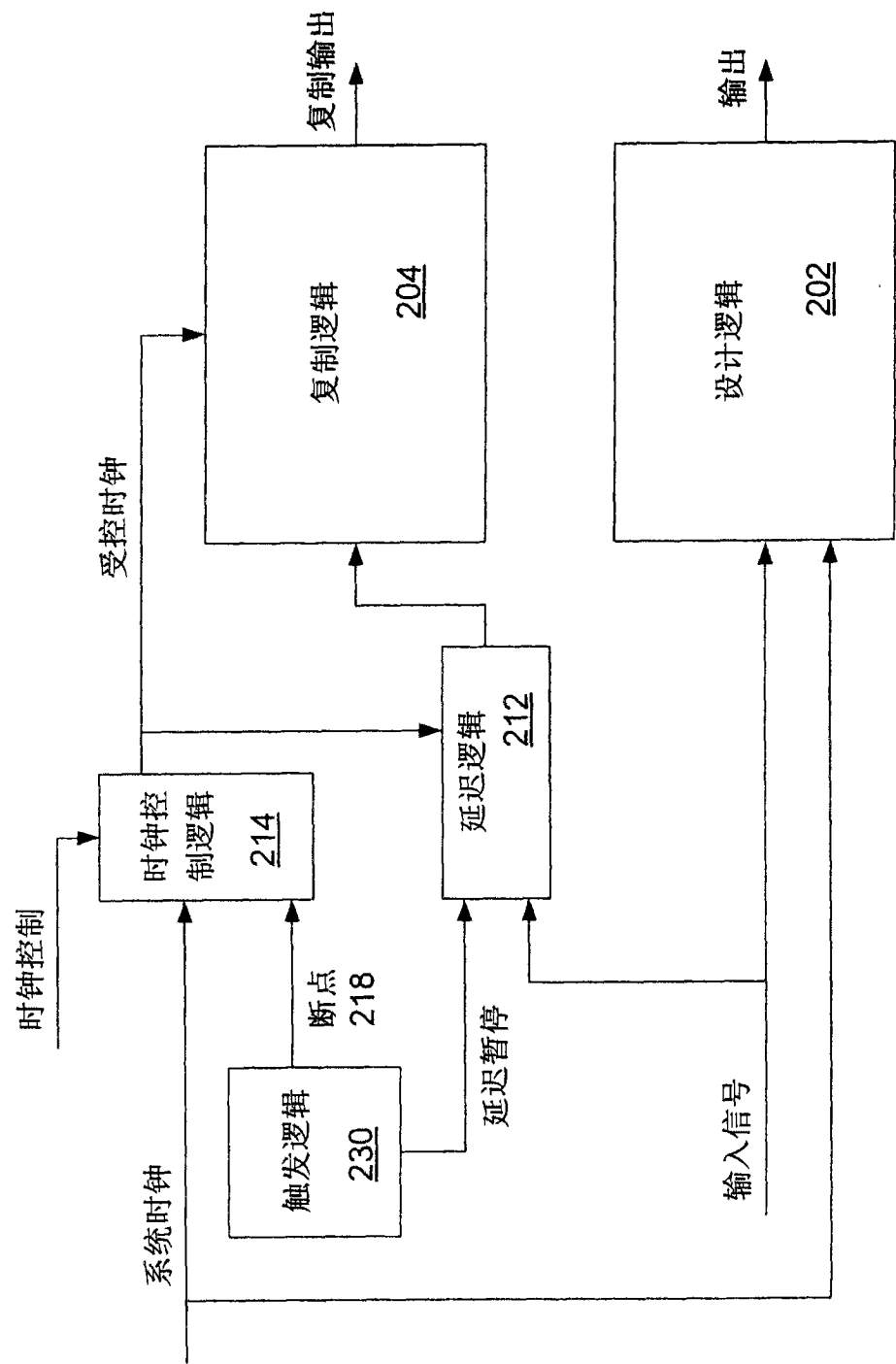


图 8

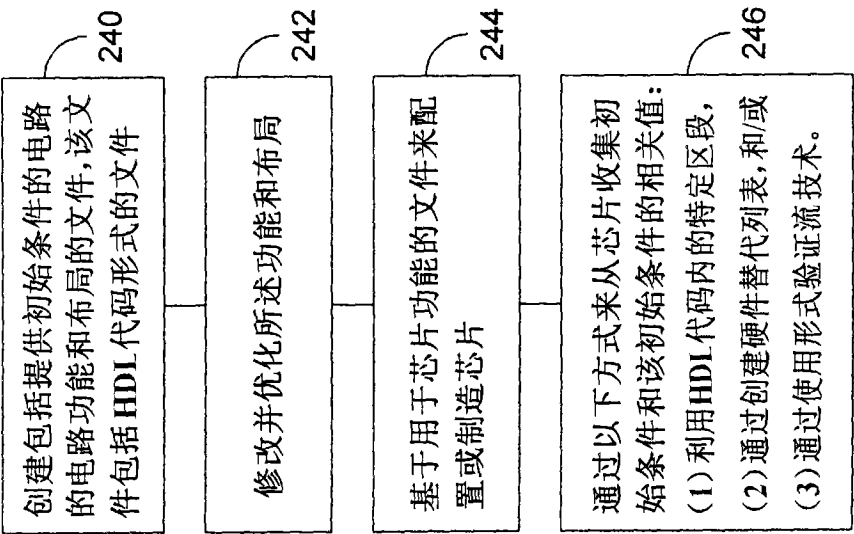


图 9

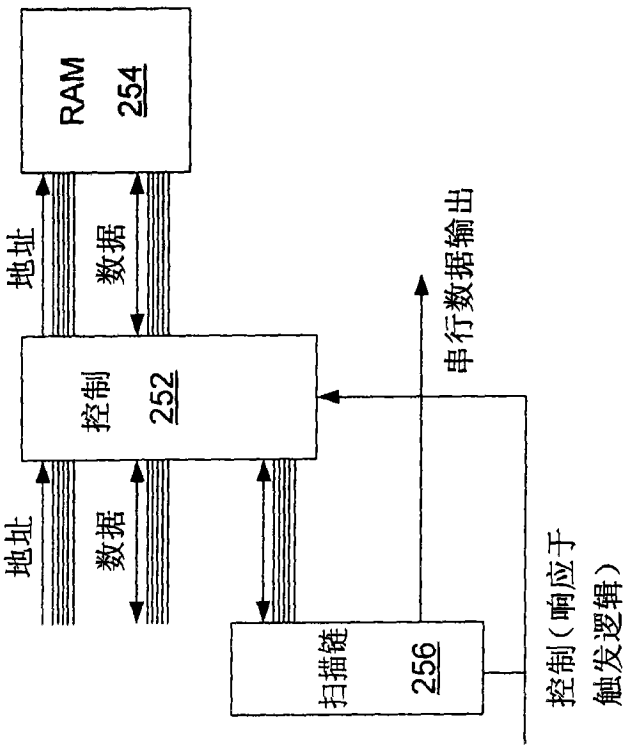


图 10

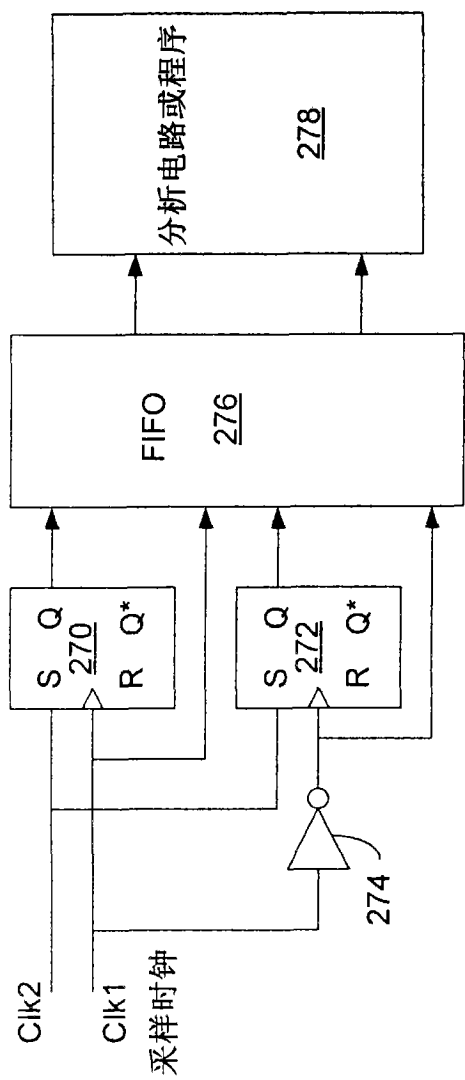


图 11

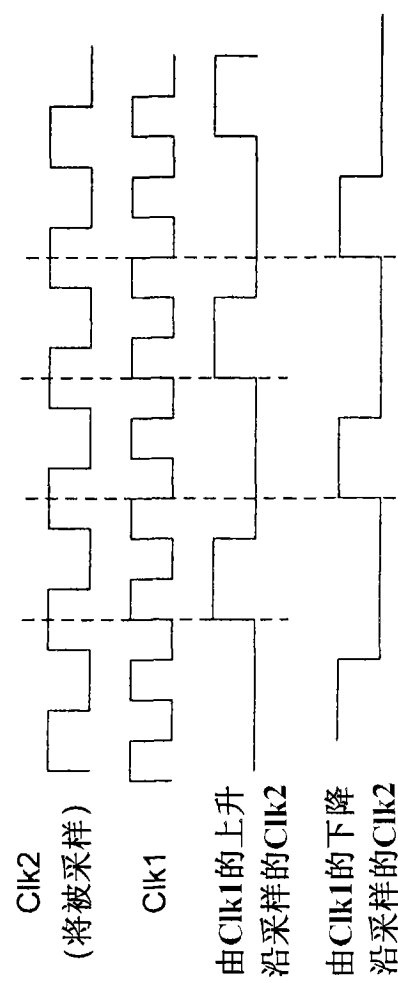


图 12

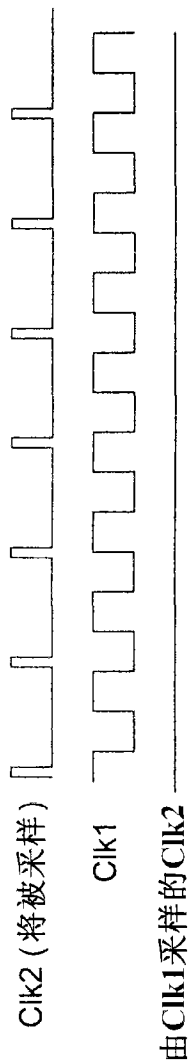


图 13

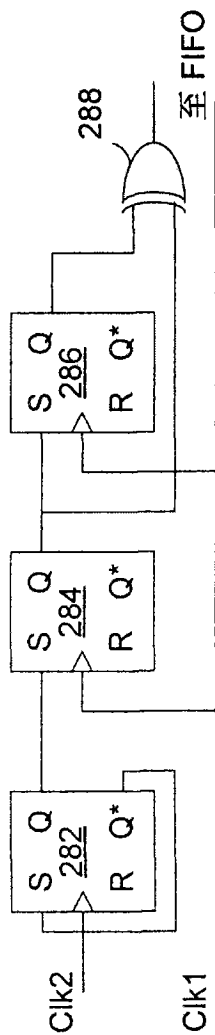


图 14

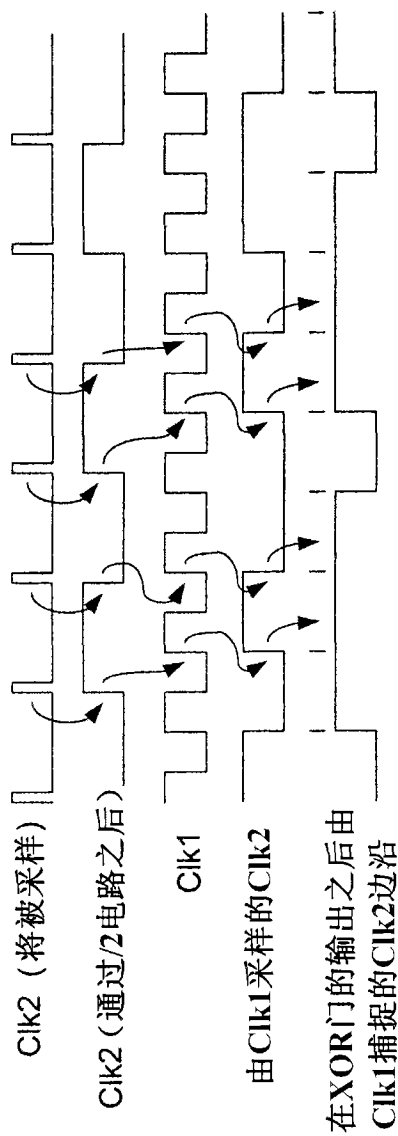


图 15

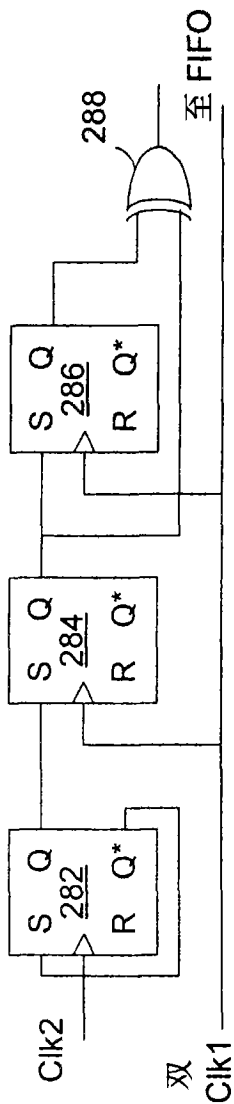


图 16

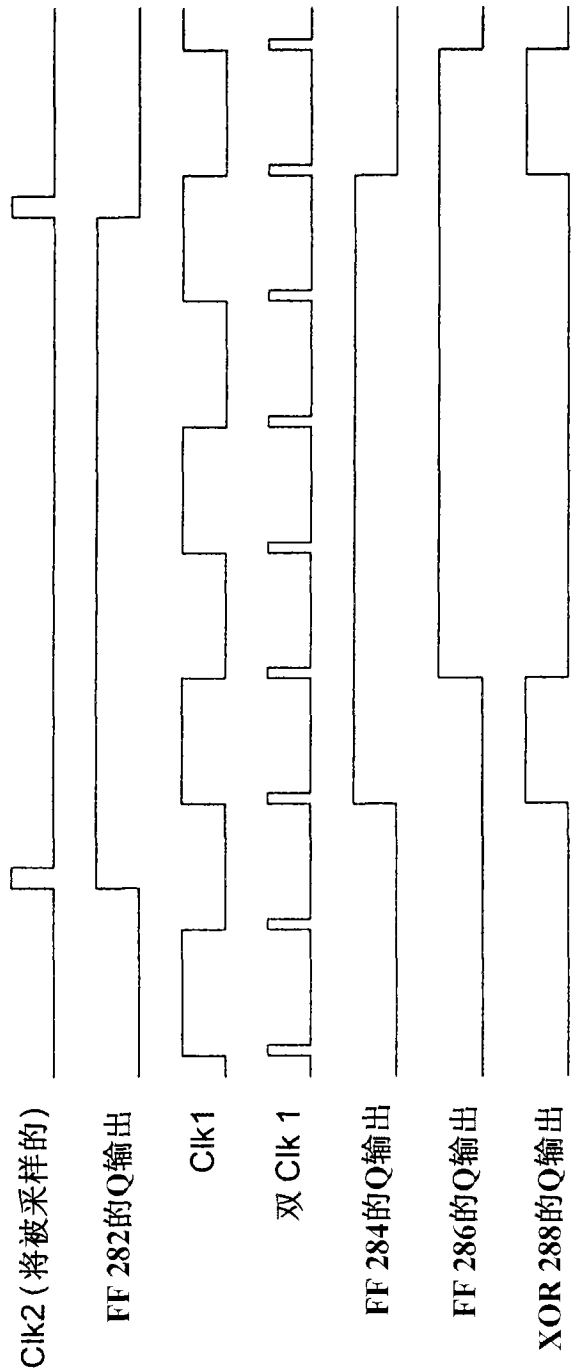


图 17

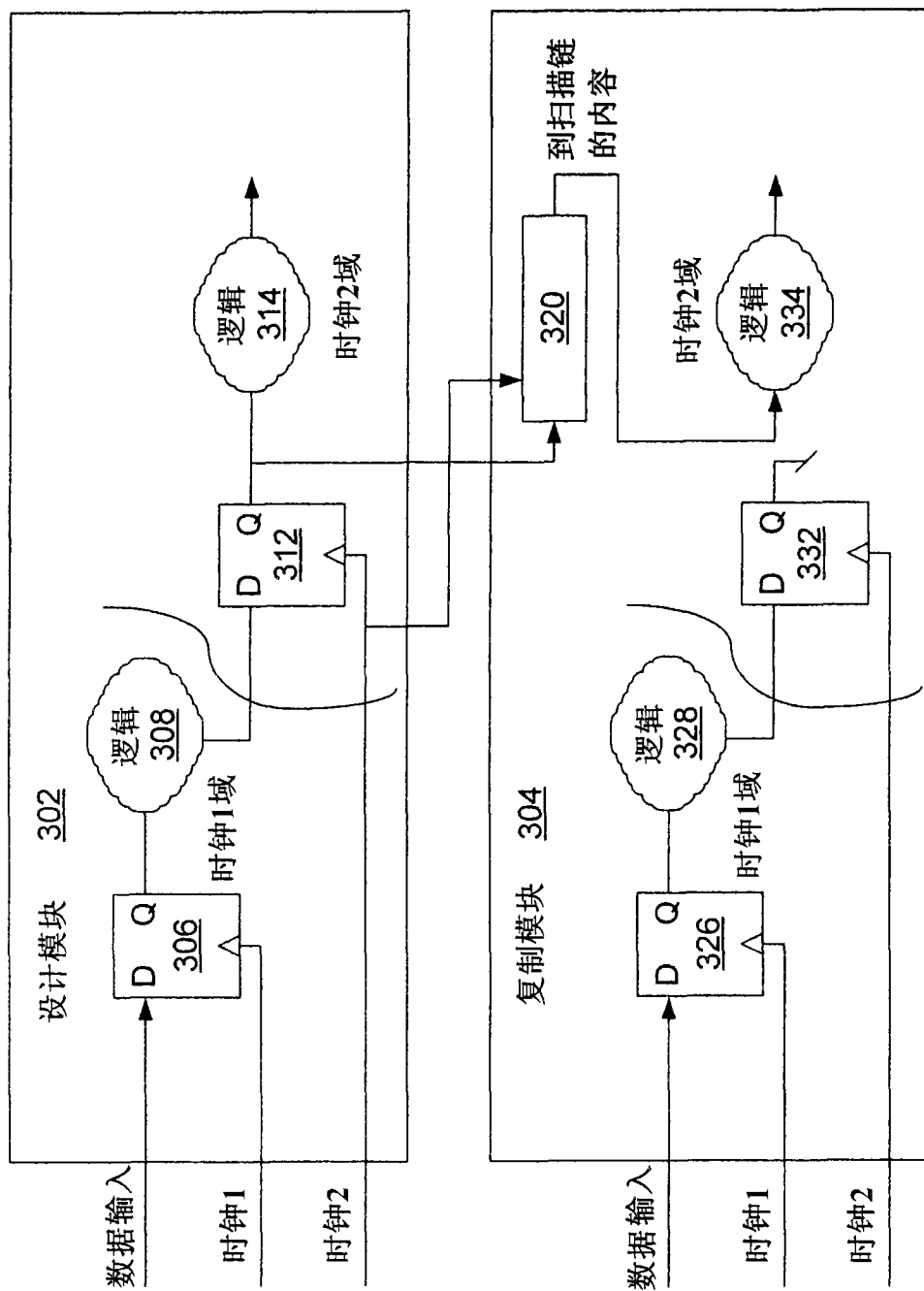


图 18

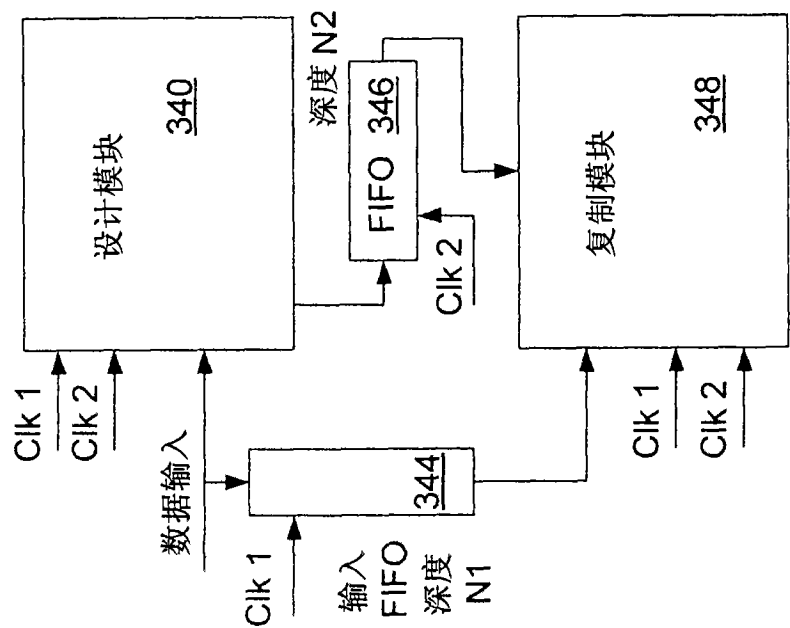


图 19

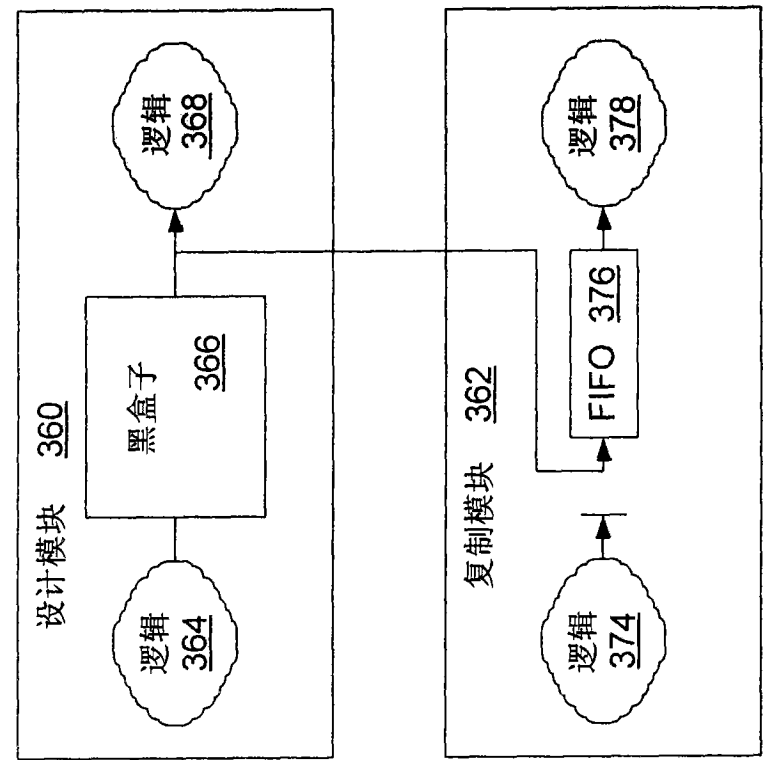


图 20

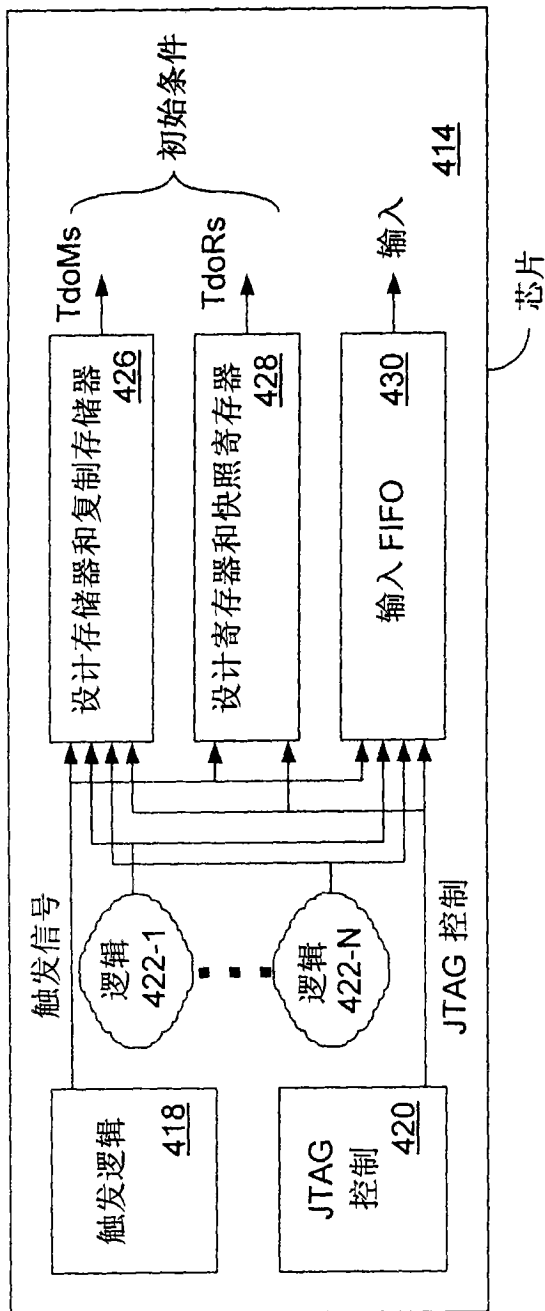


图 21

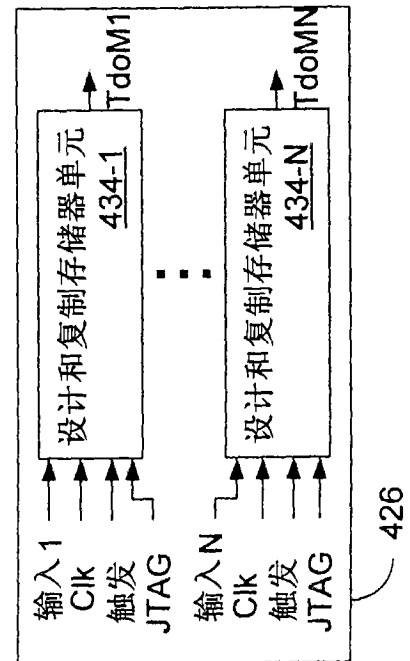


图 22

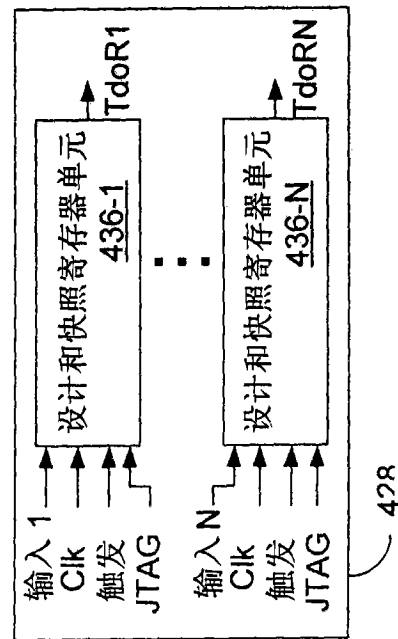


图 23

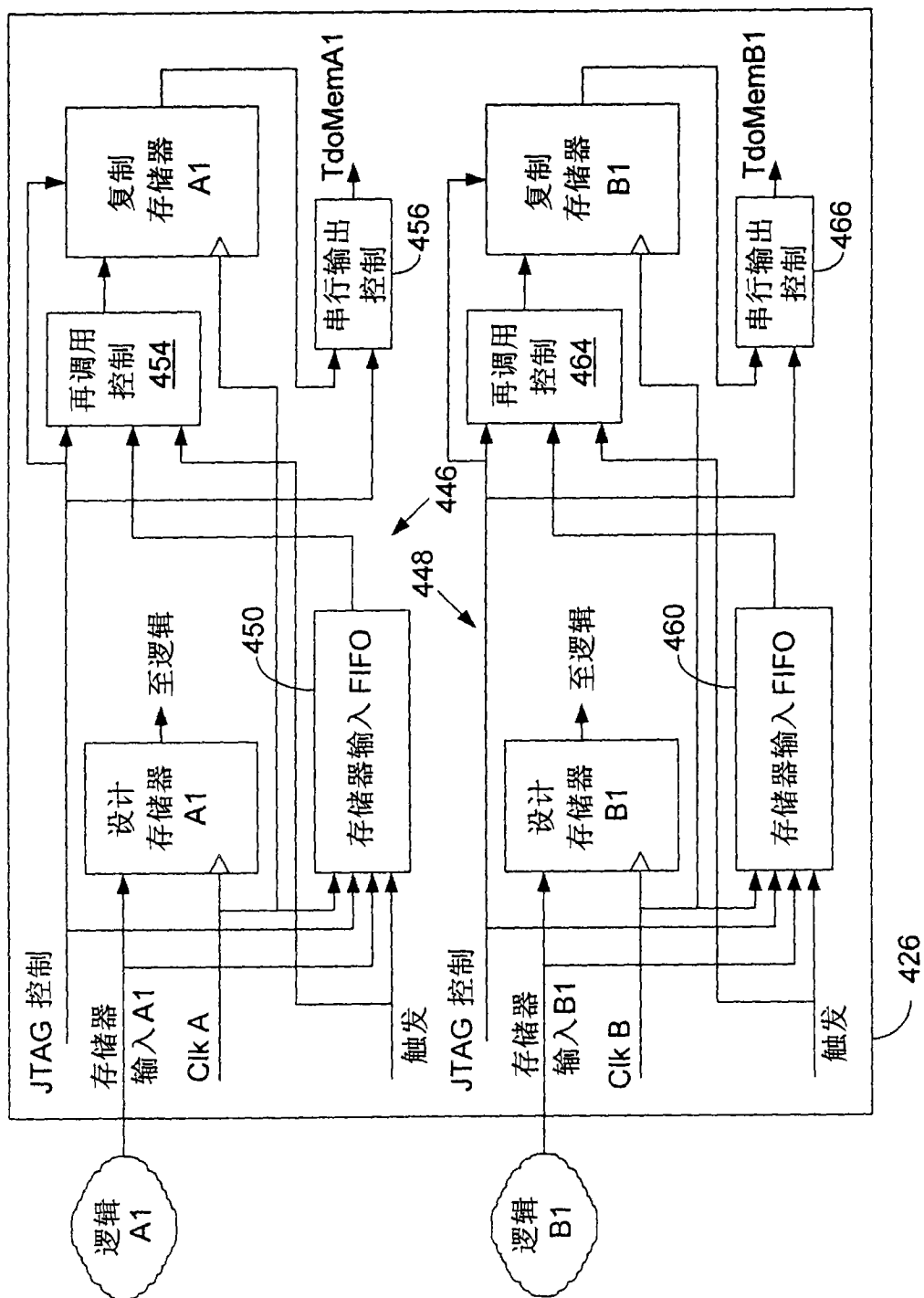


图 24

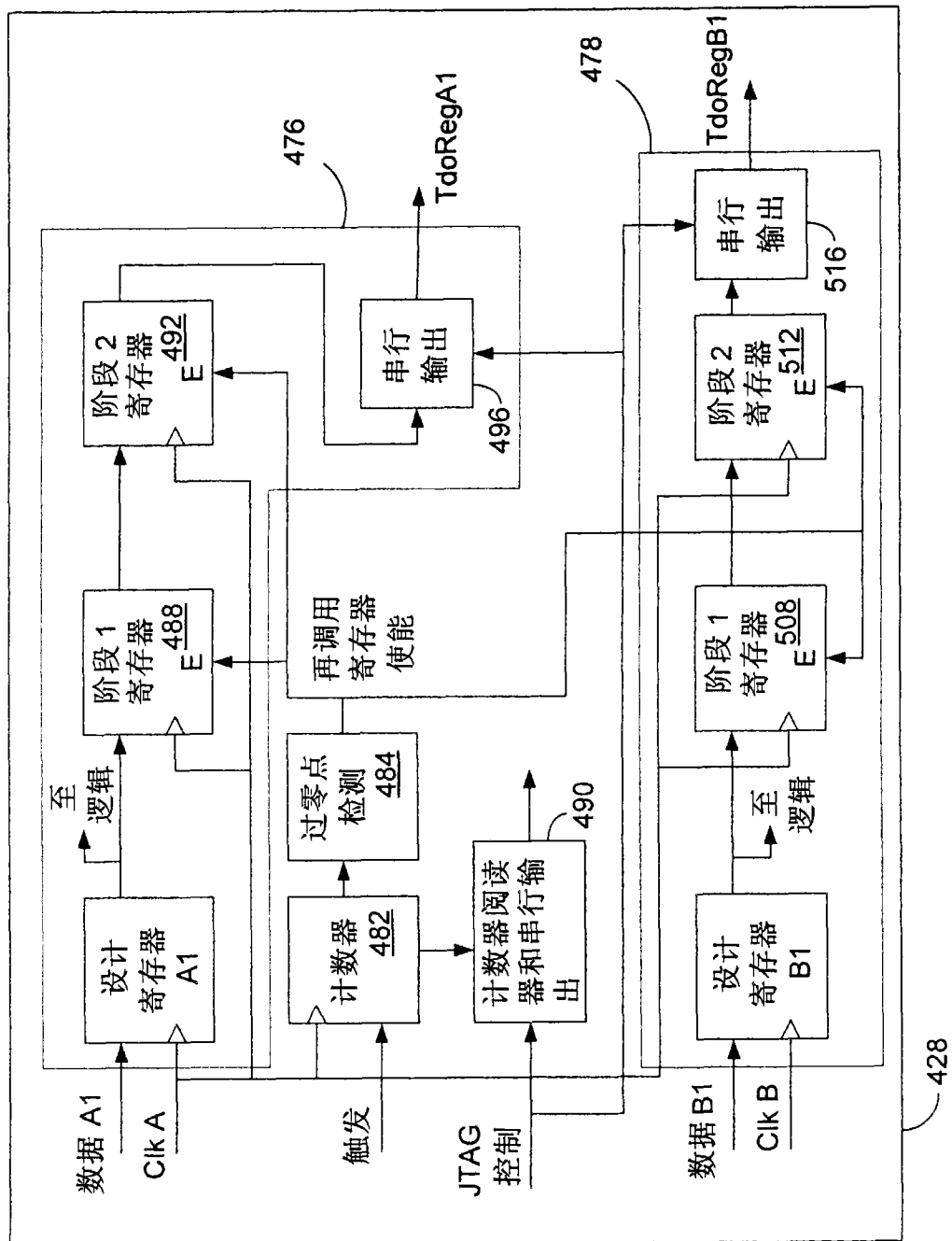


图 25

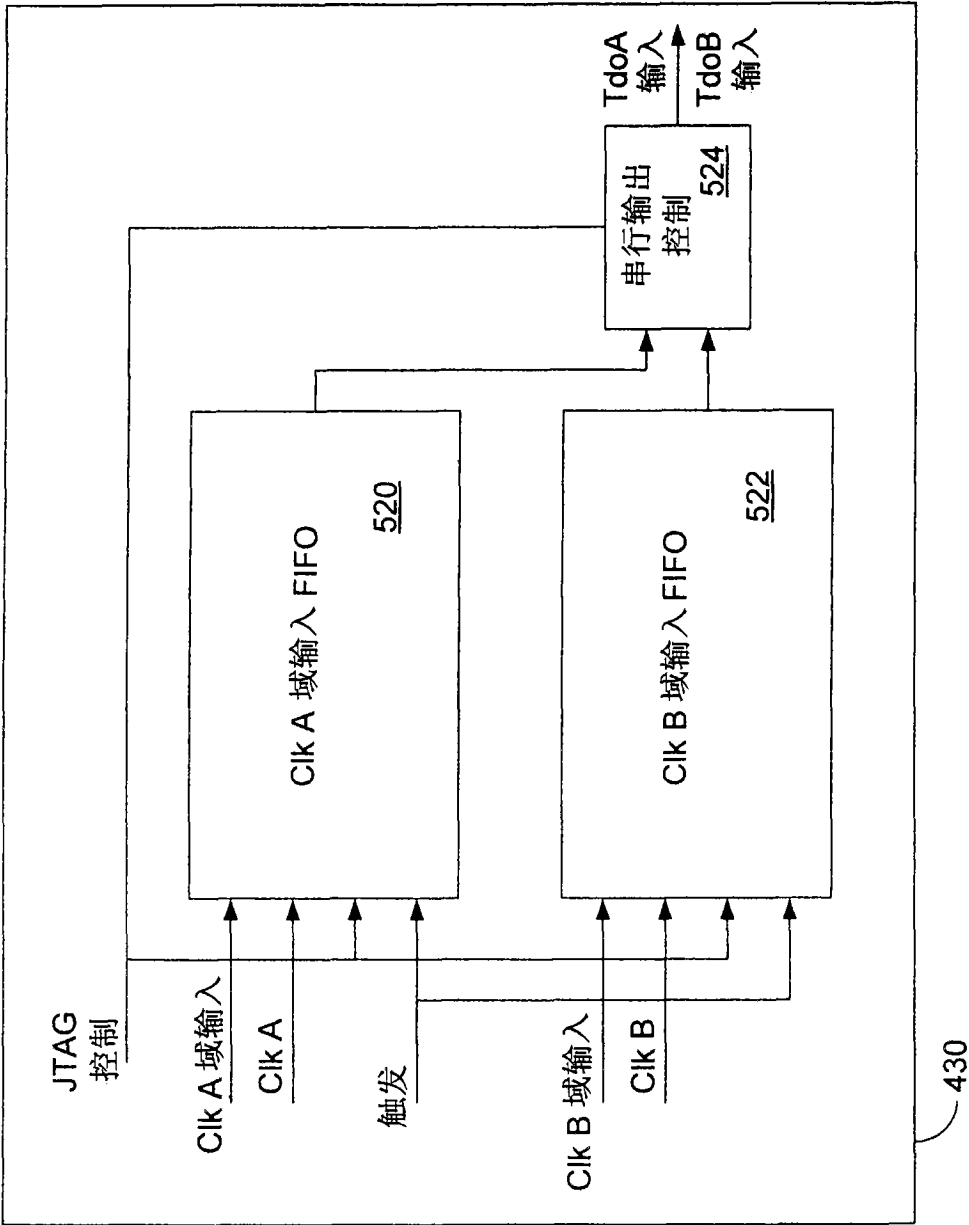


图 26

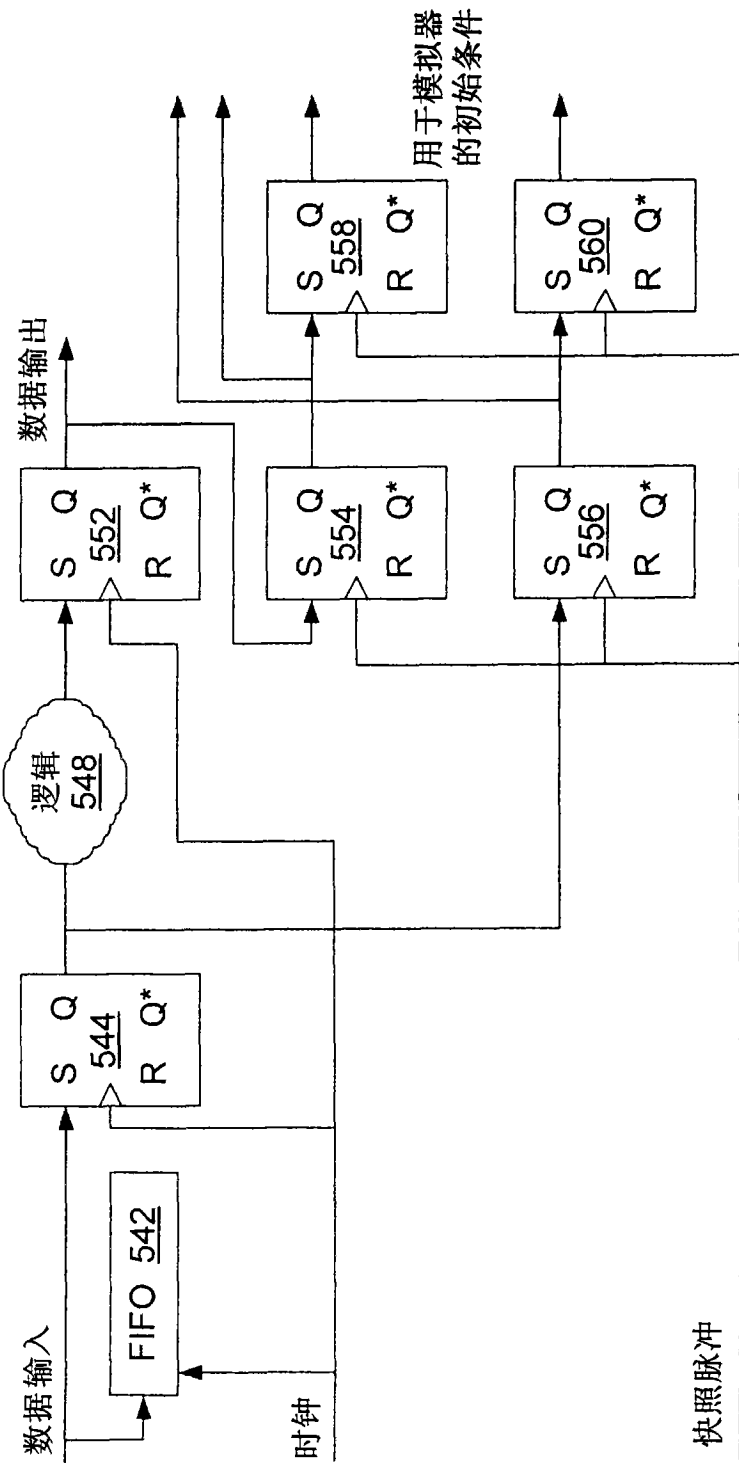


图 27

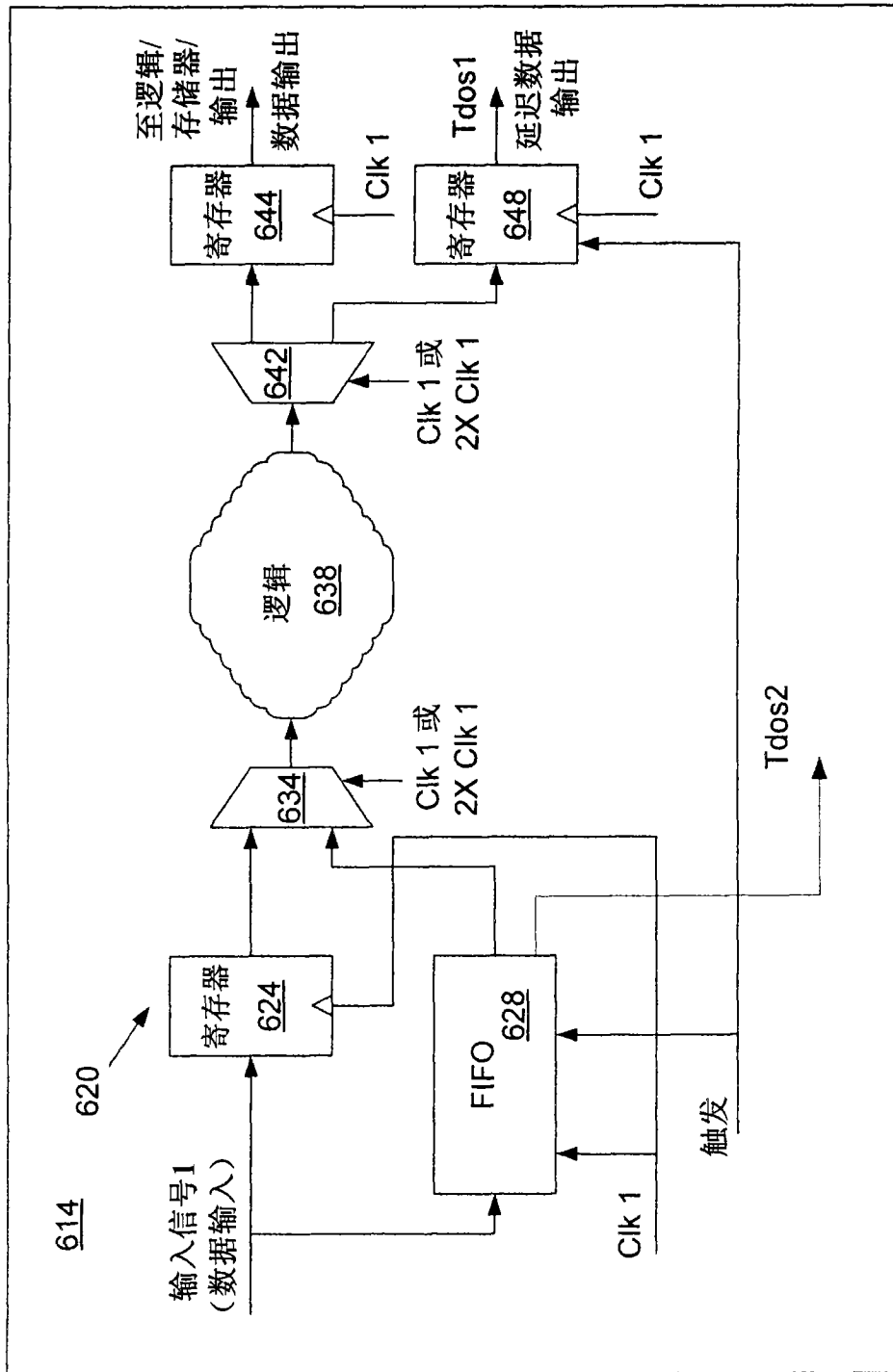


图 28

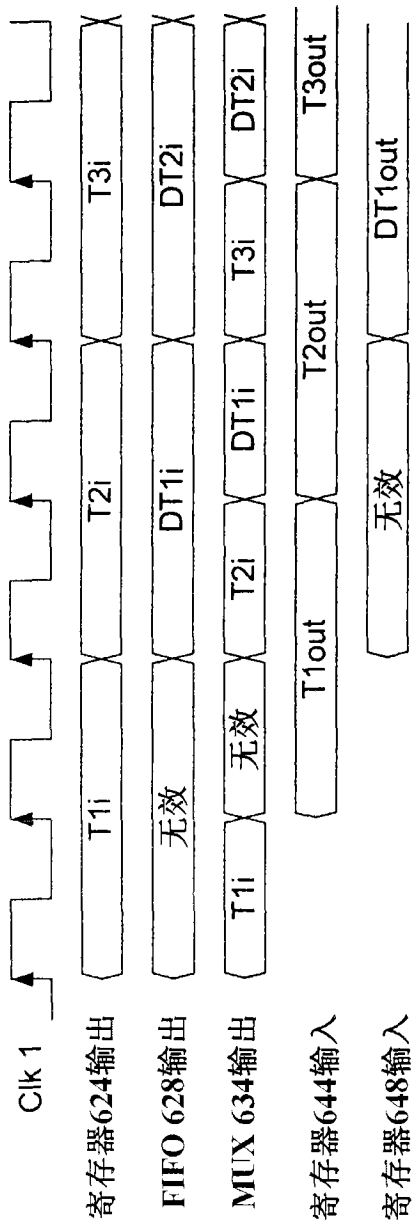


图 29

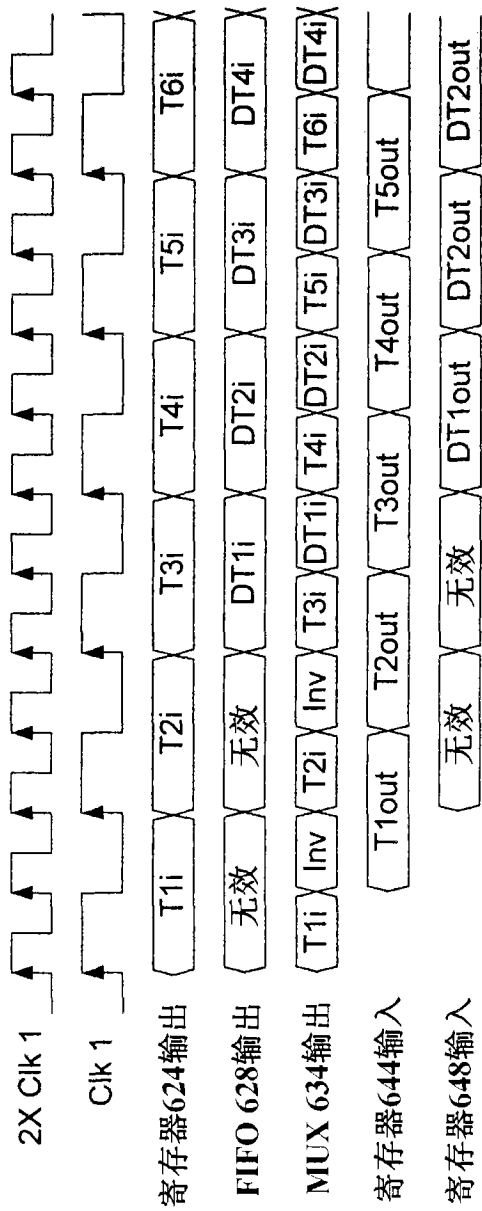


图 30

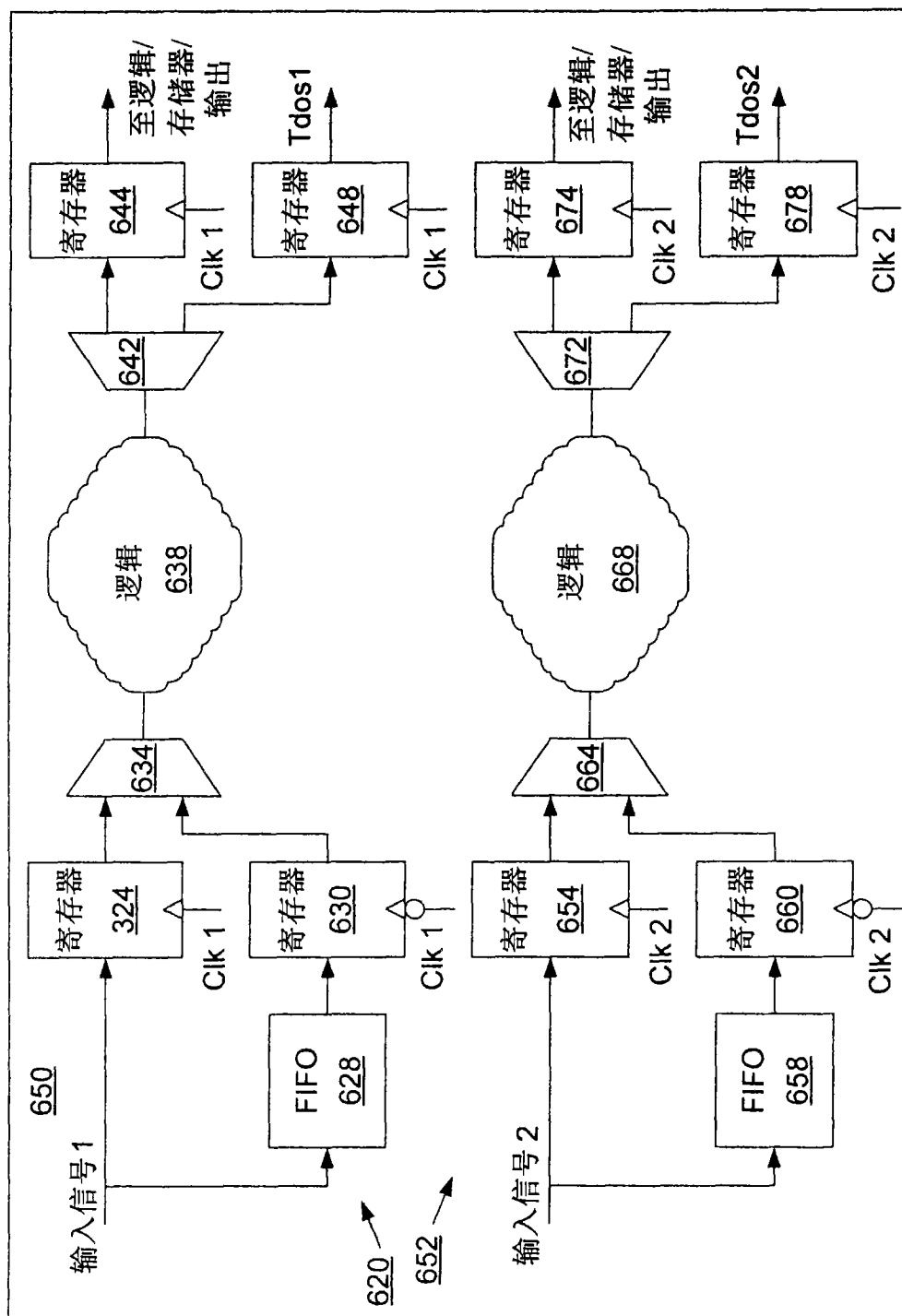


图 31

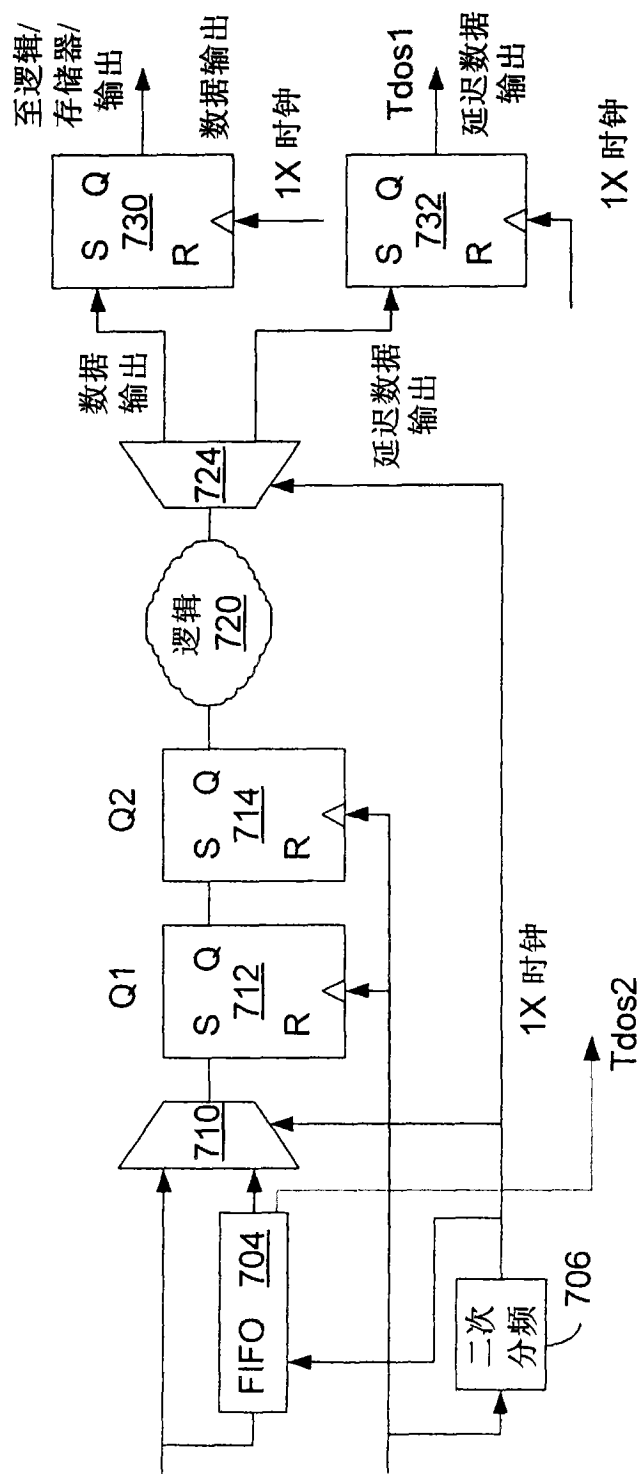


图 32

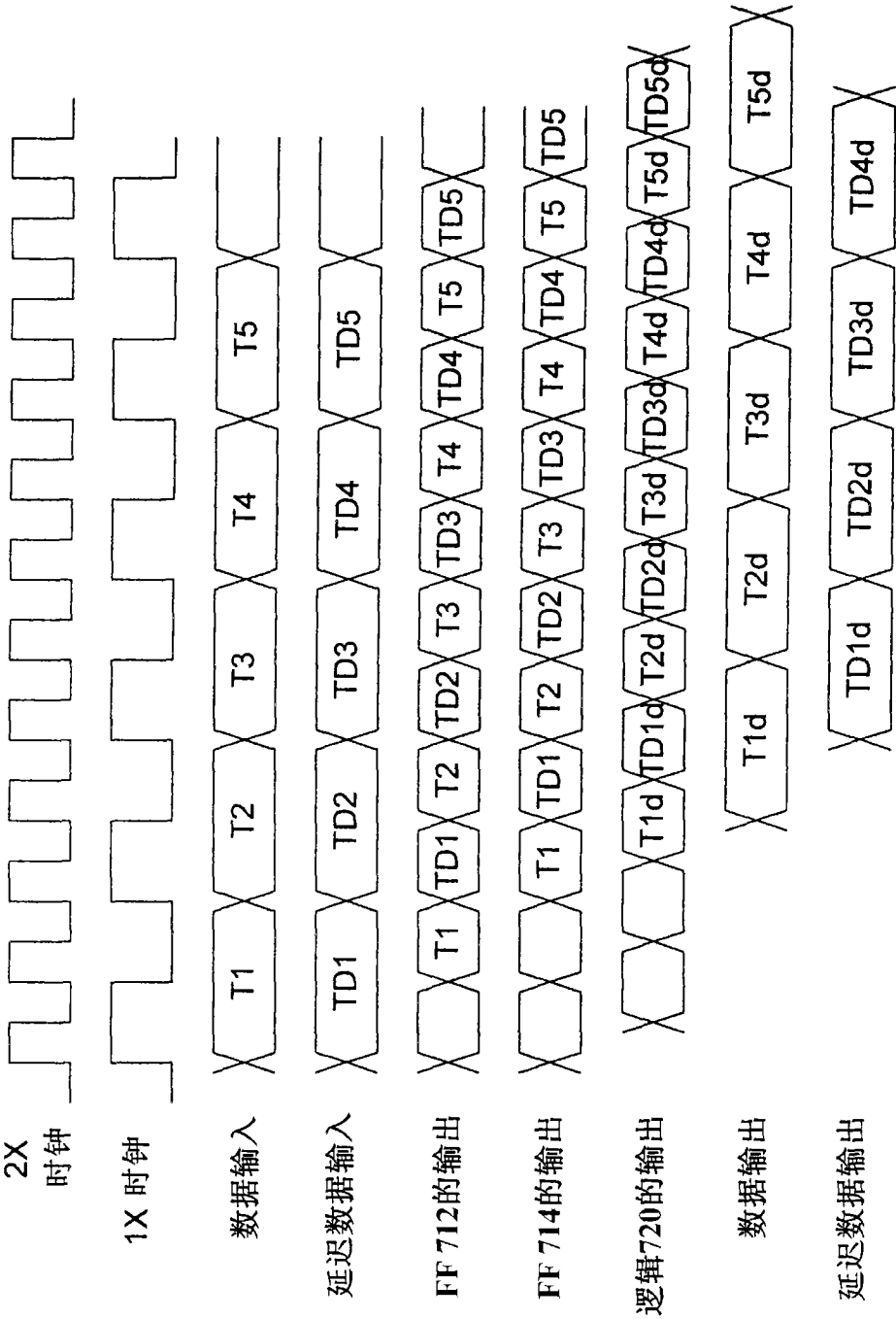


图 33

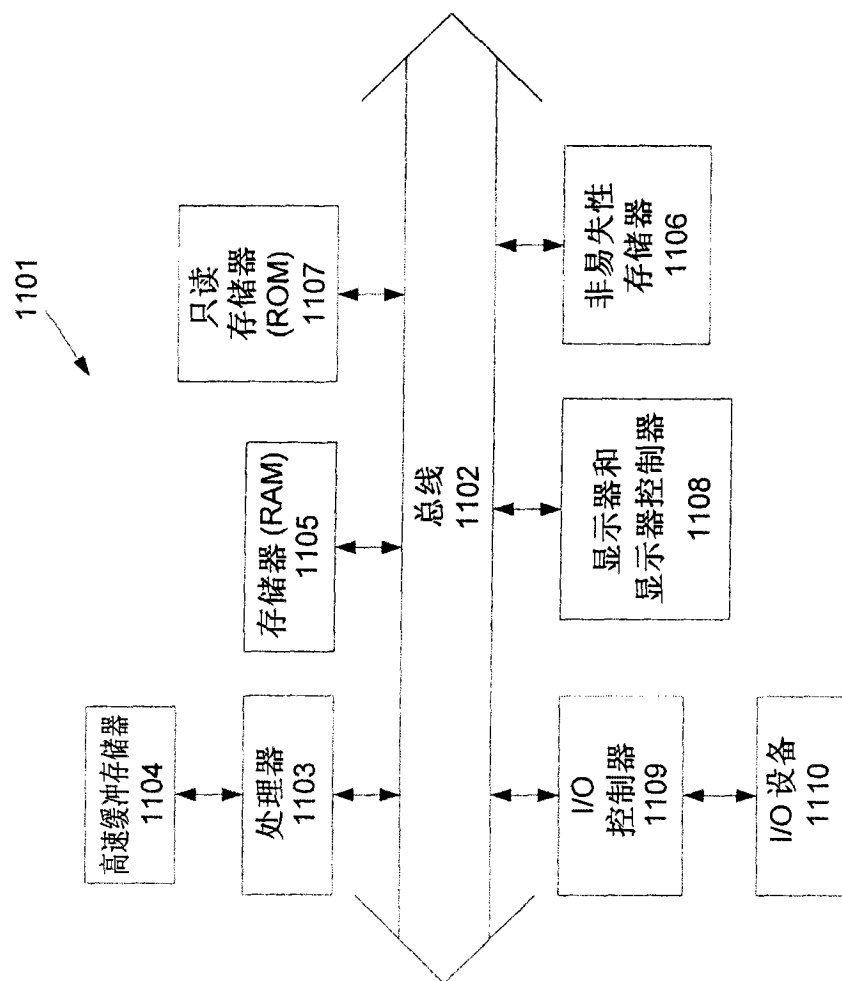


图 34