



(12) 发明专利

(10) 授权公告号 CN 102799698 B

(45) 授权公告日 2014. 07. 23

(21) 申请号 201110138733. 8

(22) 申请日 2011. 05. 26

(73) 专利权人 国际商业机器公司
地址 美国纽约

(72) 发明人 浦索明 葛亮 徐晨 于渤

(74) 专利代理机构 北京市中咨律师事务所
11247
代理人 张亚非 于静

(51) Int. Cl.
G06F 17/50 (2006. 01)

(56) 对比文件

US 2005/0278676 A1, 2005. 12. 15,
US 2009/0132975 A1, 2009. 05. 21,
CN 101351886 A, 2009. 01. 21,
CN 1732470 A, 2006. 02. 08,
蒿杰, 彭思龙. 基于超图模型的大规模门级网表层次化聚类算法. 《计算机辅助设计与图形

学报》. 2009, 第 21 卷 (第 1 期),
江立强 等. 一种有效的多时钟网络时钟树综合方案. 《计算机与数字工程》. 2005, 第 33 卷 (第 11 期),

审查员 欧晓丹

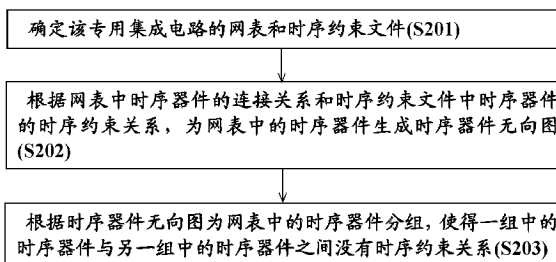
权利要求书3页 说明书9页 附图6页

(54) 发明名称

一种用于专用集成电路的时钟树规划的方法和系统

(57) 摘要

本发明公开了一种用于专用集成电路的时钟树规划的方法和系统, 该方法包括: 确定该专用集成电路的网表和时序约束文件; 根据网表中时序器件的连接关系和时序约束文件中时序器件的时序约束关系, 为网表中的时序器件生成时序器件无向图; 根据时序器件无向图为网表中的时序器件分组, 使得一组中的时序器件与另一组中的时序器件之间没有时序约束关系。使用该方法改进的专用集成电路设计方法将设计流程从周为单位减少到天, 设计人员能够迅速规划时钟树, 因此减少了设计时间, 提高了设计效率。



1. 一种用于专用集成电路的时钟树规划的方法,包括:
 - 确定该专用集成电路的网表和时序约束文件;
 - 根据网表中时序器件的连接关系和时序约束文件中时序器件的时序约束关系,为网表中的时序器件生成时序器件无向图,该步骤包括:
 - 根据网表中时序器件的连接关系,为网表中的时序器件建立连接无向图;
 - 根据时序约束文件中时序器件的时序约束关系,为网表中的时序器件建立时序无向图;以及
 - 通过合并连接无向图和时序无向图,获得时序器件无向图;
 - 根据时序器件无向图为网表中的时序器件分组,使得一组中的时序器件与另一组中的时序器件之间没有时序约束关系。
2. 根据权利要求1所述的方法,其中,所述根据网表中时序器件的连接关系,为网表中的时序器件建立连接无向图包括:
 - 在连接无向图中用节点代表网表中的时序器件;
 - 在连接无向图的节点之间建立连接边,其中,连接无向图中的一个节点与另一个节点之间的连接边,表示这两个节点所代表的时序器件在网表中是通过路径连接的;
 - 其中时序器件在网表中是否通过路径连接是根据网表中时序器件的连接关系来判断的。
3. 根据权利要求2所述的方法,其中,所述根据时序约束文件中的时序约束关系,为网表中的时序器件建立时序无向图包括:
 - 在时序无向图中用节点代表网表中的时序器件;
 - 在时序无向图的节点之间建立时序边,其中,时序无向图中的一个节点与另一个节点之间的时序边,表示这两个节点所代表的时序器件之间存在以下时序约束的至少之一:
 - 异步时钟;
 - 假性路径;
 - 多周期路径;
 - 其中,异步时钟、假性路径和多周期路径为包含在时序约束文件中的时序约束关系。
4. 根据权利要求2所述的方法,其中,所述根据时序约束文件中的时序约束关系,为网表中的时序器件建立时序无向图包括:
 - 在时序无向图中用节点代表网表中的时序器件;
 - 在时序无向图的节点之间建立时序边,其中,时序无向图中的一个节点与另一个节点之间的时序边,表示这两个节点所代表的时序器件之间存在以下时序约束的至少之一:
 - 异步时钟;
 - 假性路径;
 - 多周期路径并且两个时序器件之间的时序检查周期数超过了预设的时钟周期数;
 - 其中,异步时钟、假性路径和多周期路径为包含在时序约束文件中的时序约束关系。
5. 根据权利要求1-4之一所述的方法,其中,所述通过合并连接无向图和时序无向图,获得时序器件无向图包括:
 - 将连接无向图与时序无向图其中之一中的节点表示为时序器件无向图中的节点;
 - 在时序器件无向图的节点之间建立边,其中,时序器件无向图一个节点与另一个节

点之间的边,表示这两个节点在连接无向图中所对应的节点之间存在连接边,但是这两个节点在时序无向图中所对应的节点之间不存在时序边。

6. 根据权利要求 5 所述的方法,其中,所述根据时序器件无向图为网表中的时序器件分组包括:

将时序器件无向图中的一个连通的子图包含的节点所代表的时序器件划分在同一个组中。

7. 一种用于专用集成电路的时钟树规划的系统,包括:

确定装置,被配置为确定该专用集成电路的网表和时序约束文件;

时序器件无向图建立装置,被配置为根据网表中时序器件的连接关系和时序约束文件中时序器件的时序约束关系,为网表中的时序器件生成时序器件无向图,该装置包括:

连接无向图建立装置,被配置为根据网表中时序器件的连接关系,为网表中的时序器件建立连接无向图;

时序无向图建立装置,被配置为根据时序约束文件中时序器件的时序约束关系,为网表中的时序器件建立时序无向图;以及

合并装置,被配置为通过合并连接无向图和时序无向图,获得时序器件无向图;

分组装置,被配置为根据时序器件无向图为网表中的时序器件分组,使得一组中的时序器件与另一组中的时序器件之间没有时序约束关系。

8. 根据权利要求 7 所述的系统,其中连接无向图建立装置包括:

连接无向图节点建立装置,被配置为在连接无向图中用节点代表网表中的时序器件;

连接无向图的连接边建立装置,被配置为在连接无向图的节点之间建立连接边,其中,连接无向图中的一个节点与另一个节点之间的连接边,表示这两个节点所代表的时序器件在网表中是通过路径连接的;

其中时序器件在网表中是否通过路径连接是根据网表中时序器件的连接关系来判断的。

9. 根据权利要求 8 所述的系统,其中时序无向图建立装置包括:

时序无向图节点建立装置,被配置为在时序无向图中用节点代表网表中的时序器件;

时序无向图的时序边建立装置,被配置为在时序无向图的节点之间建立时序边,其中,时序无向图中的一个节点与另一个节点之间的时序边,表示这两个节点所代表的时序器件之间存在以下时序约束的至少之一:

异步时钟;

假性路径;

多周期路径;

其中异步时钟、假性路径和多周期路径为包含在时序约束文件中的时序约束关系。

10. 根据权利要求 8 所述的系统,其中时序无向图建立装置包括:

时序无向图节点建立装置,被配置为在时序无向图中用节点代表网表中的时序器件;

时序无向图的时序边建立装置,被配置为在时序无向图的节点之间建立时序边,其中,时序无向图中的一个节点与另一个节点之间的时序边,表示这两个节点所代表的时序器件之间存在以下时序约束的至少之一:

异步时钟;

假性路径；

多周期路径并且两个时序器件之间的时序检查周期数超过了预设的时钟周期数；

其中异步时钟、假性路径和多周期路径为包含在时序约束文件中的时序约束关系。

11. 根据权利要求 7-10 之一所述的系统,其中合并装置包括：

时序器件无向图节点建立装置,被配置为将连接无向图与时序无向图其中之一中的节点表示为时序器件无向图中的节点；

时序器件无向图边建立装置,被配置为在时序器件无向图的节点之间建立边,其中,时序器件无向图中一个节点与另一个节点之间的边,表示这两个节点在连接无向图中所对应的节点之间存在连接边,但是这两个节点在时序无向图中所对应的节点之间不存在时序边。

12. 根据权利要求 11 所述的系统,其中,所述分组装置进一步配置成将时序器件无向图中的一个连通的子图包含的节点所代表的时序器件划分在同一个组中。

一种用于专用集成电路的时钟树规划的方法和系统

技术领域

[0001] 本发明一般涉及集成电路设计,更具体地,涉及一种用于专用集成电路的时钟树规划的方法和系统。

背景技术

[0002] 在集成电路技术领域中,专用集成电路(ASIC)是指应特定用户要求和特定电子系统的需要而设计、制造的集成电路。ASIC的特点是面向特定用户的需求,在批量生产时与通用集成电路相比具有体积更小、功耗更低、可靠性提高、性能提高、保密性增强、成本降低等优点。

[0003] 图1示出了现有的专用集成电路设计的流程,在步骤S101,生成门级网表(Gate Level Netlist),在步骤S102,进行布局(Full Placement)。在步骤S103,执行静态时序分析(Static Timing Analysis),该步骤对于时钟效应的预计采用理想时钟模型(Ideal Clock Model)的方式。在步骤S104,插入时钟树(Insert Clock Tree),也称时钟树规划。现有的时钟树规划过程可以手工完成,也可以由电子设计自动化(EDA)工具来完成的,最流行的时钟树都采用自动化工具来完成。其基本原理是根据各时序器件到时钟源的距离,在适当的位置放置缓冲器(buffer)以使时钟源到每个时序器件的时钟偏差(skew)最小化。插入的由缓冲器组成的树状的信号中继网络被称为时钟树。插入的时钟树要求能够满足时序约束文件中的时序约束,从而保持时序的收敛性,达到设计的效果。然后在步骤S105,进行带时钟的时序分析以及时钟树调节,在步骤S106,进行步线以及布线后时序收敛分析,即时钟平衡;最后在步骤S107,生成版图。这样,就完成了初步的设计流程。

[0004] 然而,上述图1的设计流程在现代超大规模专用集成电路的设计中遇到了一些问题,主要体现在:

[0005] 随着专用集成电路的规模越来越大,时钟树规划这一工作本身也越来越耗时。对于千万门级的超大规模专用集成电路来讲,现有的设计工具采用的设计原则是使得所有的时钟树的终端(也就是时序器件,例如寄存器,SRAM等)的时钟偏差和时钟延迟尽可能小,例如Clock Designer, Encounter, ICC等设计工具就是依照这样的设计原则,对每一个时序器件进行时钟平衡,使得时钟树规划、时钟树插入以及时钟树调节的耗时往往要以天甚至周来计算,造成设计周期长,产品不能很快上市。

[0006] 在诸如Clock Designer, Encounter, ICC等设计工具还存在和上述实施方式相结合的另一时钟树的规划方式,该实施方式中电路设计人员通过与逻辑设计人员沟通,将所有的时序器件进行分组,然后在每一组时序器件内部进行时钟平衡,最后考虑各组间的时钟平衡。由于时序器件的数量非常大,这种沟通需要反复迭代,设计周期依然很长。

[0007] 上述两个现有的解决方案都会造成产品设计周期过长,可能错过了有利的市场时机。

发明内容

[0008] 因此,需要一种方法,该方法使得设计人员能够迅速规划时钟树,从而减少设计时间。本发明提出了一种用于专用集成电路的时钟树规划的方法和系统,其中对时序器件分组,分组后,没有时序约束的器件被分在不同组内,减少同一个时钟树根节点连接的时序器件的数量,最大程度减少时钟树内的冗余平衡,降低时钟树平衡的复杂度,这样在组内就可以使用现有的设计工具中采用的使得所有的时钟树的终端(也就是时序器件)的时钟偏差和时钟延迟尽可能小的设计原则插入时钟树,达到时钟平衡。使用该方法改进的专用集成电路设计方法和系统可以将设计流程从周为单位减少到天,设计人员能够迅速规划时钟树,因此减少了设计时间,提高了设计效率。

[0009] 根据本发明的一个方面,提供了一种用于专用集成电路的时钟树规划的方法,包括:

[0010] 确定该专用集成电路的网表和时序约束文件;

[0011] 根据网表中时序器件的连接关系和时序约束文件中时序器件的时序约束关系,为网表中的时序器件生成时序器件无向图;

[0012] 根据时序器件无向图为网表中的时序器件分组,使得一组中的时序器件与另一组中的时序器件之间没有时序约束关系。

[0013] 根据本发明的另一个方面,提供了一种用于专用集成电路的时钟树规划的系统,包括:

[0014] 确定装置,被配置为确定该专用集成电路的网表和时序约束文件;

[0015] 时序器件无向图建立装置,被配置为根据网表中时序器件的连接关系和时序约束文件中时序器件的时序约束关系,为网表中的时序器件生成时序器件无向图;

[0016] 分组装置,被配置为根据时序器件无向图为网表中的时序器件分组,使得一组中的时序器件与另一组中的时序器件之间没有时序约束关系。

附图说明

[0017] 通过对附图中本发明实施例实施方式的更详细描述,本发明的上述、以及其它目的、特征和优势将变得更加明显,其中,相同的参考标号通常代表本发明实施例实施方式中的相同部件。

[0018] 图 1 示出了现有的专用集成电路设计的流程;

[0019] 图 2 示意性地示出了根据本发明实施例的一种用于专用集成电路的时钟树规划的方法步骤;

[0020] 图 3 示出了图 2 中步骤 S202 的具体流程;

[0021] 图 4 示出了一个示例性的专用集成电路;

[0022] 图 5 示出了异步时钟的定义;

[0023] 图 6 示出了假性路径的定义;

[0024] 图 7 示出了多周期路径的定义;

[0025] 图 8 示出了为图 4 所示的专用集成电路建立的连接无向图、时序无向图以及时序器件无向图;

[0026] 图 9 示出了对另外一个未示出网表以及时序约束文件的专用集成电路,建立的连接无向图、时序无向图以及时序器件无向图;以及

[0027] 图 10 示出了一种用于专用集成电路的时钟树规划的系统的结构。

具体实施方式

[0028] 将参照附图更加详细地描述本发明的优选实施方式,在附图中显示了本发明的优选实施例。然而,本发明可以以各种形式实现而不应该理解为被这里阐述的实施例所限制。相反,提供这些实施例是为了使本发明更加透彻和完整,并且,完全将本发明的范围传达给本领域的技术人员。

[0029] 首先明确一些基本概念以便后面发明的描述。

[0030] 网表:表达数字电路的器件的拓扑连接关系的文件或者数据结构。

[0031] 时钟树:插入的由缓冲器组成的树状的信号中继网络被称为时钟树。因此,一个专用集成电路一般有一个或多个时钟树。

[0032] 时钟树的延迟时间(latency):是指对于一个单独的时钟树,时钟信号从根节点到叶节点的传递时间为时钟树的延迟时间。

[0033] 时钟树的偏差(Skew):是指对于一个单独的时钟树上的各个时序器件,时钟树的延迟时间的不同。

[0034] 时钟树效应:本发明中是指时钟树的时序效应,主要包括时钟树的延迟,和时钟树的偏差。

[0035] 在现有的专用集成电路设计领域,逻辑设计人员与电路设计人员是两组不同的设计人员,逻辑设计人员了解电路的功能及要求,而电路设计人员只是根据逻辑设计人员的设计要求进行电路设计,他们并不具体了解电路的功能。在业界,由逻辑设计人员交给电路设计人员的标准交付件包括网表和时序约束文件。然后电路设计人员完成图 1 的步骤 S102 到步骤 S107 的设计过程。时序设计要求设计的电路满足时序约束文件中的时序要求,例如,对于一个时钟树中的所有时序器件,时钟树的延迟是相同的,也就是时钟信号基本同时达到各个时序器件的时钟管脚,如果有些时序器件延迟大,有些时序器件延迟小,就在延迟小的时序器件的时序路径中插入缓冲器,增大延迟,最后使得时钟信号基本同时达到各个时序器件的时钟管脚,时钟树的偏差最小,达到时钟树平衡。另外有一些时序器件之间不存在时序约束,就无需平衡。

[0036] 对于一个时序器件很多的复杂设计,在本发明提供的时钟树规划方法(即图 1 的步骤 S104)中,可以进一步分为两个步骤:首先对时序器件分组,使得每组时序器件之间不存在时序约束,只有组内的时序器件之间存在时序约束;然后对每一组时序器件,采用现有的时钟树规划方法满足时序平衡。

[0037] 在本发明的一种实施方式中,提供了一种用于专用集成电路的时钟树规划的方法,可以使得每组时序器件之间不存在时序约束,只有组内的时序器件之间存在时序约束,并且电路设计人员无需与逻辑设计人员迭代沟通,只需要获得标准交付件,即可执行。图 2 示意性地示出了该方法的步骤,并且结合一个实例,来陈述该方法步骤。根据图 2,在步骤 S201,确定该专用集成电路的网表和时序约束文件。图 4 示出了一个示例性的专用集成电路,图 4 中,左边的 Clk 信号表示时钟信号的源端,Data 1 和 Data 2 表示数据输出,Data 3 和 Data 4 表示数据输入,每个 REG 表示一个时序器件,AND 表示与门,OR 表示一个或门。图 4 的专用集成电路的网表如下。该网表至少包含构成该时钟树的源器件 Clk 和时序器件

REG 及其连接关系。这里,时钟树的源端为所述构成该时钟树的源器件的时钟输入端,时钟树的终端为所述构成该时钟树的源器件的时钟输出端。

[0038] 图 4 所示的专用集成电路的网表的内容如下:

[0039]

```
input    Data1, Data2, Clk1, Clk2;  
output  Data3, Data4;  
wire Net1, Net2, Net3, Net4, Net5, Net6, Net7;  
...  
REG REG1( //定义第一时序器件的连接关系;  
    .D (Data1),    //第一时序器件的管脚 D 与 Data1 相连;  
    .E (Clk1),    //第一时序器件的管脚 E 与 Clk1 相连;  
    .L2(Net1)     //第一时序器件的管脚 L2 与 Net1 相连;
```

[0040]


```

);
REG REG2( //定义第二时序器件的连接关系;
.D (Net3),
.E (Clk1),
.L2(Net4)
);
REG REG3( //定义第三时序器件的连接关系;
.D (Net4),
.E (Clk2),
.L2(Net5)
);
REG REG4(
.D (Net5),
.E (Clk2),
.L2(Data3)
);
REG REG5(
.D (Data2),
.E (Clk2),
.L2(Net7)
);
REG REG6(
.D (Net7),
.E (Clk2),
.L2(Data4)
);
AND AND1( //定义第一与门的连接关系;
.A(NET1),
.B(VDD),
.Y(NET2)
);
OR OR1( //定义第一或门的连接关系;
.A(NET2),
.B(NET1),
.Y(NET3)
);

```

[0041] 图 4 所示的专用集成电路的时序约束文件的内容如下：

[0042] Timing Constraint：

[0043] create_clock CLK1[get_ports Clk1]-period 10-waveform{05}# 产生一个时钟，周期为 10，上升沿为 0，下降沿在 5

[0044] create_clock CLK2[get_ports Clk2]-period 14-waveform{07}

[0045] set_clock_uncertainty0.1[get_clocks CLK1]# 设置时钟边沿抖动

```
[0046] set_clock_uncertainty0.1[get_clocks CLK2]
[0047] set_input_delay-clock CLK1 4[get_ports {Data1}]# 设置信号的输入延迟
[0048] set_input_delay-clock CLK2 3[get_ports {Data2}]
[0049] set_output_delay 1[get_ports {Data3}]# 设置信号的输出延迟
[0050] set_output_delay 1[get_ports {Data4}]
[0051] set_clock_group-asynchronous-name g1-group CLK1-group CLK2
[0052] # 设置 CLK1 和 CLK2 之间为异步时钟
[0053] set_false_path-from REG5/L2-to REG6/D # 设置两点路径为假性约束路径
[0054] set_multicycle_path 2-from REG1/L2-to REG2/D # 设置路径为多周期延迟路
```

[0055] 返回图 2, 在步骤 S202, 根据网表中时序器件的连接关系和时序约束文件中时序器件的时序约束关系, 为网表中的时序器件生成时序器件无向图。图 3 示出了图 2 中步骤 S202 的具体流程, 根据图 3, 在步骤 S301, 根据网表中时序器件的连接关系, 为网表中的时序器件建立连接无向图; 在步骤 S302, 根据时序约束文件中时序器件的时序约束关系, 为网表中的时序器件建立时序无向图; 在步骤 S303, 通过合并连接无向图和时序无向图, 获得时序器件无向图。

[0056] 在建立连接无向图的过程中, 首先要获得网表中的时序器件。数字电路的每一个器件 (包括 IP), 都有表示是否时序器件的标记, 作为器件的特征之一, 列举在设计的库文件中。因此, 在获得时钟树相关的网表之后, 根据某器件的类型, 逐一查对库文件, 就可以判断该器件是否为时序器件。这样, 就可以提取所述时钟树连接的所有时序器件。另外一种实施方式中, 可以使用某种数据结构 (例如表、数组等) 列出网表中所涉及的所有时序器件的名称, 然后用程序排查网表中的所有器件的名称是否在该数据结构所列出的名称范围之内, 从而获得所有时序器件。然后在连接无向图中用节点代表网表中的时序器件; 最后在连接无向图的节点之间建立连接边, 其中, 连接无向图中的一个节点与另一个节点之间的连接边, 表示这两个节点所代表的时序器件在网表中是通过路径连接的。这里, 时序器件在网表中是否通过路径连接是根据网表中时序器件的连接关系来判断的。上述路径包括连接路径或者组合路径。这里, 连接路径指两个时序器件直接相连; 组合路径包含器件路径和分支路径, 其中, 器件路径指两个时序器件之间通过至少一个非时序器件相连, 分支路径指两个时序器件相连的部分存在电路分支。也就是说, 两个时序器件之间只要有一条路径相通, 无论该条路径中间包含多少器件和分支路径, 都在两个时序器件对应的节点之间建立一条连接边。由于在网表中包含了所有器件的连接关系, 通过这些连接关系, 就可以判断任意两个时序器件之间是否通过路径相连。

[0057] 时序无向图的建立比较复杂。时序无向图中, 仍然将网表中的时序器件表示为一个节点, 在时序约束文件中, 只要符合下列三种情况之一, 就在两个时序器件之间建立一条时序边。

[0058] (1) 异步时钟 (Asynchronous Clocks): 图 5 示出了异步时钟的定义, 两个时序器件有各自驱动的时钟, 无法预知两个时钟的上升或下降沿之间的关系, 则时序无向图中在这两个时序器件之间建立一条时序边。

[0059] (2) 假性路径 (False Path): 图 6 示出了假性路径的原理, 两个时序器件有路径,

但是它们之间不需要做时序约束,称为假性路径。时序无向图中,在存在假性路径的时序器件之间建立一条时序边。

[0060] (3) 多周期路径 (Multicycle path with enough margin):通常两个时序器件之间的时序检查周期数为一个时钟周期,当两个时序器件之间的时序检查周期数超过一个时钟周期的约束时,称为多周期路径,图 7 示出了多周期路径的定义。时序无向图中,可以在存在多周期路径的时序器件之间建立一条时序边。

[0061] 应当可以理解的是,在上述三种情况下,时钟都不需要平衡。

[0062] 因此,在时序无向图的节点之间建立时序边时,时序无向图中的一个节点与另一个节点之间的如果建立了时序边,就表示这两个节点所代表的时序器件之间存在上述时序约束的至少之一。建立时序边的时序器件之间不需要时钟平衡。这里,异步时钟、假性路径和多周期路径为包含在时序约束文件中的时序约束关系,换句话说,时序约束文件中包含了很多时序约束关系,异步时钟、假性路径和多周期路径是其中的一种约束关系类型,但是并不表示时序约束文件中都包含这种时序约束关系,可能时序约束文件中不包含任何上述三种时序约束关系的一种,这时,时序无向图中就只有节点,不存在时序边。

[0063] 在进一步的实施方式中,对多周期路径可以设定一个参数,该参数表示两个时序器件之间的时序检查周期数超过该设定的时钟周期数才在两个时序器件之间建立一条时序边。这时,时序无向图的建立过程就包括:首先在时序无向图中用节点代表网表中的时序器件;然后在时序无向图的节点之间建立时序边,其中,时序无向图中的一个节点与另一个节点之间的时序边,表示这两个节点所代表的时序器件之间存在以下时序约束的至少之一:异步时钟;假性路径;多周期路径并且两个时序器件之间的时序检查周期数超过了预设的时钟周期数;其中异步时钟、假性路径和多周期路径为包含在时序约束文件中的时序约束关系。如果两个节点所代表的时序器件之间不存在上述时序约束关系,两个节点之间就不建立时序边。

[0064] 在合并连接无向图和时序无向图过程中,首先将连接无向图与时序无向图其中之一中的节点表示为时序器件无向图中的节点;由上述两个无向图的建立过程可见,这两种无向图包含的节点是相同的。然后在时序器件无向图的节点之间建立边,其中,时序器件无向图中一个节点与另一个节点之间的边,表示这两个节点在连接无向图中所对应的节点之间存在连接边,但是这两个节点在时序无向图中所对应的节点之间不存在时序边。这样就完成了无向图合并。也就是说,在合并的时序器件无向图中,对于存在两条边的节点,即既有连接边又有时序边,删除二者之间的边;但是只存在连接边,不存在时序边,则保留二者之间的边。由于建立了时序边的时序器件之间不需要时钟平衡,因此,在时序器件无向图中,建立了时序边的时序器件之间无边。根据上述方法的步骤 S301, S302 和 S303 对图 4 所示的集成电路建立的连接无向图、时序无向图以及时序器件无向图如图 8 所示。对另外一个未示出网表以及时序约束文件的专用集成电路,建立的连接无向图、时序无向图以及时序器件无向图如图 9 所示。

[0065] 返回图 2,在步骤 S203,根据时序器件无向图为网表中的时序器件分组,使得一组中的时序器件与另一组中的时序器件之间没有时序约束关系。具体来说,就是将时序器件无向图中的一个连通的子图包含的节点所代表的时序器件划分在同一个组中。一个单独的节点可以作为一个连通子图,因此其对应的时序器件也分成一组。例如,图 8 的时序器件

无向图包含 5 个连通子图,就可以将这些时序器件分成 5 个组,后续的时序分析中,只需要对包含两个时序器件的组进行时钟平衡,时钟平衡由原来的 6 个时序器件减少为两个时序器件;图 9 的时序器件无向图包含 6 个连通子图,可以将这些时序器件分成 6 个组,后续的时序分析只需要对包含两个或两个以上时序器件的组进行时钟平衡,时钟平衡由原来的 19 个时序器件减少 4 个组,每组分别为 7、5、3、2 个时序器件。后续的时钟平衡工作将大大减少。

[0066] 使用上述方法分组后可以使得各组之间的时序器件之间不存在时序关系,所以无需时钟平衡;并且每组内部时序器件数量可以大大减少,在组内使用现有的设计工具中采用的使得所有的时钟树的终端(也就是时序器件)的时钟偏差和时钟延迟尽可能小的设计原则插入时钟树,达到时钟平衡。使用该方法改进的专用集成电路设计方法将设计流程从周为单位减少到天,设计人员能够迅速规划时钟树,因此减少了设计时间,提高了设计效率。

[0067] 在同一个发明构思下,本发明还公开了一种用于专用集成电路的时钟树规划的系统 1000,包括:确定装置 1001,被配置为确定该专用集成电路的网表和时序约束文件;时序器件无向图建立装置 1002,被配置为根据网表中时序器件的连接关系和时序约束文件中时序器件的时序约束关系,为网表中的时序器件生成时序器件无向图;以及分组装置 1003,被配置为根据时序器件无向图将网表中的时序器件分组,使得一组中的时序器件与另一组中的时序器件之间没有时序约束关系。

[0068] 在一种实施方式中,时序器件无向图建立装置包括(图 10 未示出):连接无向图建立装置,被配置为根据网表中时序器件的连接关系,为网表中的时序器件建立连接无向图;时序无向图建立装置,被配置为根据时序约束文件中时序器件的时序约束关系,为网表中的时序器件建立时序无向图;以及合并装置,被配置为通过合并连接无向图和时序无向图,获得时序器件无向图。

[0069] 在另一种实施方式中,连接无向图建立装置包括(图 10 未示出):连接无向图节点建立装置,被配置为在连接无向图中用节点代表网表中的时序器件;连接无向图的连接边建立装置,被配置为在连接无向图的节点之间建立连接边,其中,连接无向图中的一个节点与另一个节点之间的连接边,表示这两个节点所代表的时序器件在网表中是通过路径连接的;其中时序器件在网表中是否通过路径连接是根据网表中时序器件的连接关系来判断的。

[0070] 在又一种实施方式中,时序无向图建立装置包括(图 10 未示出):时序无向图节点建立装置,被配置为在时序无向图中用节点代表网表中的时序器件;时序无向图的时序边建立装置,被配置为在时序无向图的节点之间建立时序边,其中,时序无向图中的一个节点与另一个节点之间的时序边,表示这两个节点所代表的时序器件之间存在以下时序约束的至少之一:异步时钟;假性路径;多周期路径;其中异步时钟、假性路径和多周期路径为包含在时序约束文件中的时序约束关系。

[0071] 在再一种实施方式中,时序无向图建立装置包括(图 10 未示出):时序无向图节点建立装置,被配置为在时序无向图的节点之间建立时序边,其中,时序无向图中的一个节点与另一个节点之间的时序边,表示这两个节点所代表的时序器件之间存在以下时序约束的至少之一:异步时钟;假性路径;多周期路径并且两个时序器件之间的路径超过预设的

时钟周期个数；其中异步时钟、假性路径和多周期路径为包含在时序约束文件中的时序约束关系。

[0072] 在又一种实施方式中，合并装置包括（图 10 未示出）：时序器件无向图节点建立装置，被配置为将连接无向图与时序无向图其中之一中的节点表示为时序器件无向图中的节点；时序器件无向图边建立装置，被配置为在时序器件无向图的节点之间建立边，其中，时序器件无向图中一个节点与另一个节点之间的边，表示这两个节点在连接无向图中所对应的节点之间存在连接边，但是这两个节点在时序无向图中所对应的节点之间不存在时序边。

[0073] 在再一种实施方式中，分组装置进一步配置成将时序器件无向图中的一个连通的子图包含的节点所代表的时序器件划分在同一个组中。

[0074] 本发明可以采取硬件实施方式、软件实施方式或既包含硬件组件又包含软件组件的实施方式的形式。在优选实施方式中，本发明实现为软件，其包括但不限于固件、驻留软件、微代码等。

[0075] 而且，本发明还可以采取可从计算机可用或计算机可读介质访问的计算机程序产品的形式，这些介质提供程序代码以供计算机或任何指令执行系统使用或与其结合使用。出于描述目的，计算机可用或计算机可读机制可以是任何有形的装置，其可以包含、存储、通信、传播或传输程序以由指令执行系统、装置或设备使用或与其结合使用。

[0076] 介质可以是电的、磁的、光的、电磁的、红外线的、或半导体的系统（或装置或器件）或传播介质。计算机可读介质的例子包括半导体或固态存储器、磁带、可移动计算机磁盘、随机访问存储器（RAM）、只读存储器（ROM）、硬磁盘和光盘。目前光盘的例子包括紧凑盘 - 只读存储器（CD-ROM）、压缩盘 - 读 / 写（CD-R/W）和 DVD。

[0077] 适合于存储 / 或执行程序代码的数据处理系统将包括至少一个处理器，其直接地或通过系统总线间接地耦合到存储器元件。存储器元件可以包括在程序代码的实际执行期间所利用的本地存储器、大容量存储器、以及提供至少一部分程序代码的临时存储以便减少执行期间从大容量存储器必须取回代码的次数的快速缓存存储器。

[0078] 输入 / 输出或 I/O 设备（包括但不限于键盘、显示器、指点设备等等）可以直接地或通过中间 I/O 控制器耦合到系统。

[0079] 网络适配器也可以耦合到系统，以使得数据处理系统能够通过中间的私有或公共网络而耦合到其他数据处理系统或远程打印机或存储设备。调制解调器、线缆调制解调器以及以太网卡仅仅是当前可用的网络适配器类型的几个例子。

[0080] 从上述描述应当理解，在不脱离本发明真实精神的情况下，可以对本发明各实施方式进行修改和变更。本说明书中的描述仅仅是用于说明性的，而不应被认为是限制性的。本发明的范围仅受所附权利要求书的限制。



图 1

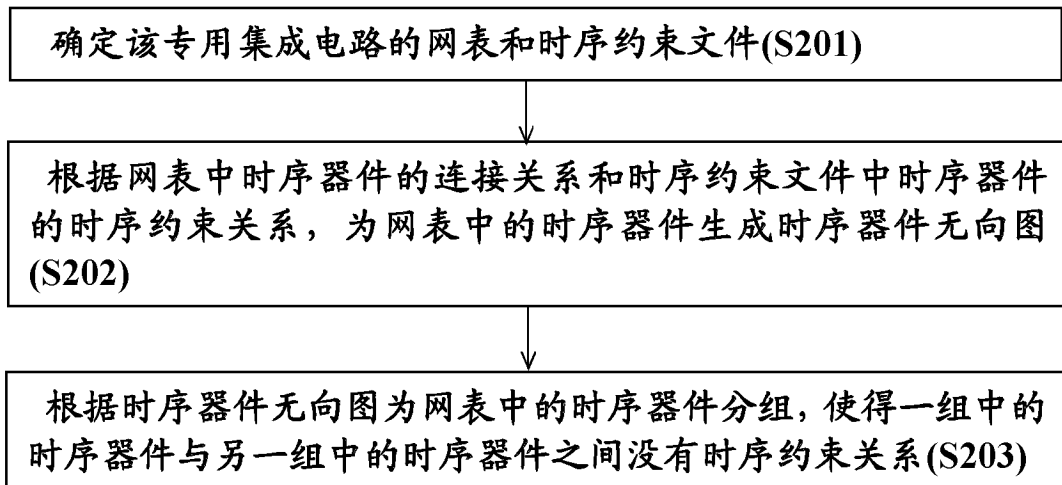


图 2

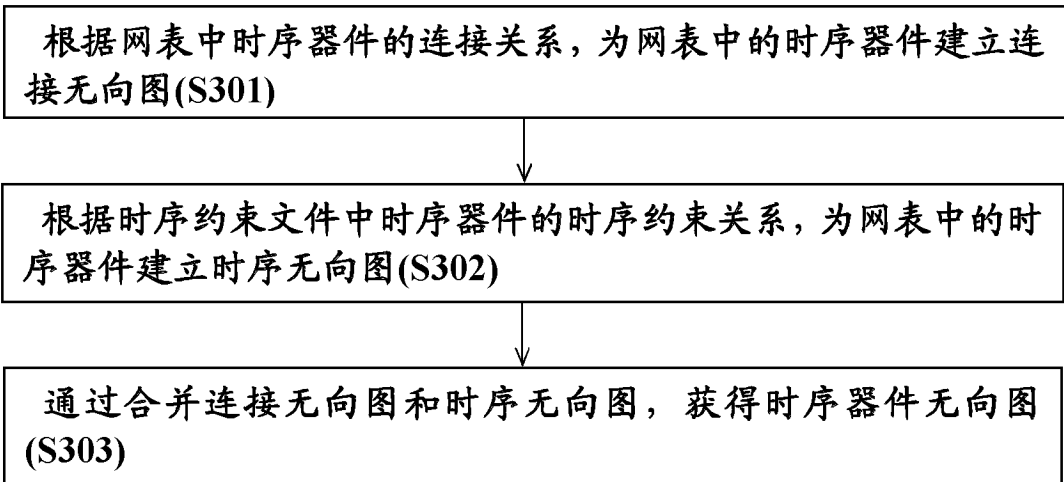


图 3

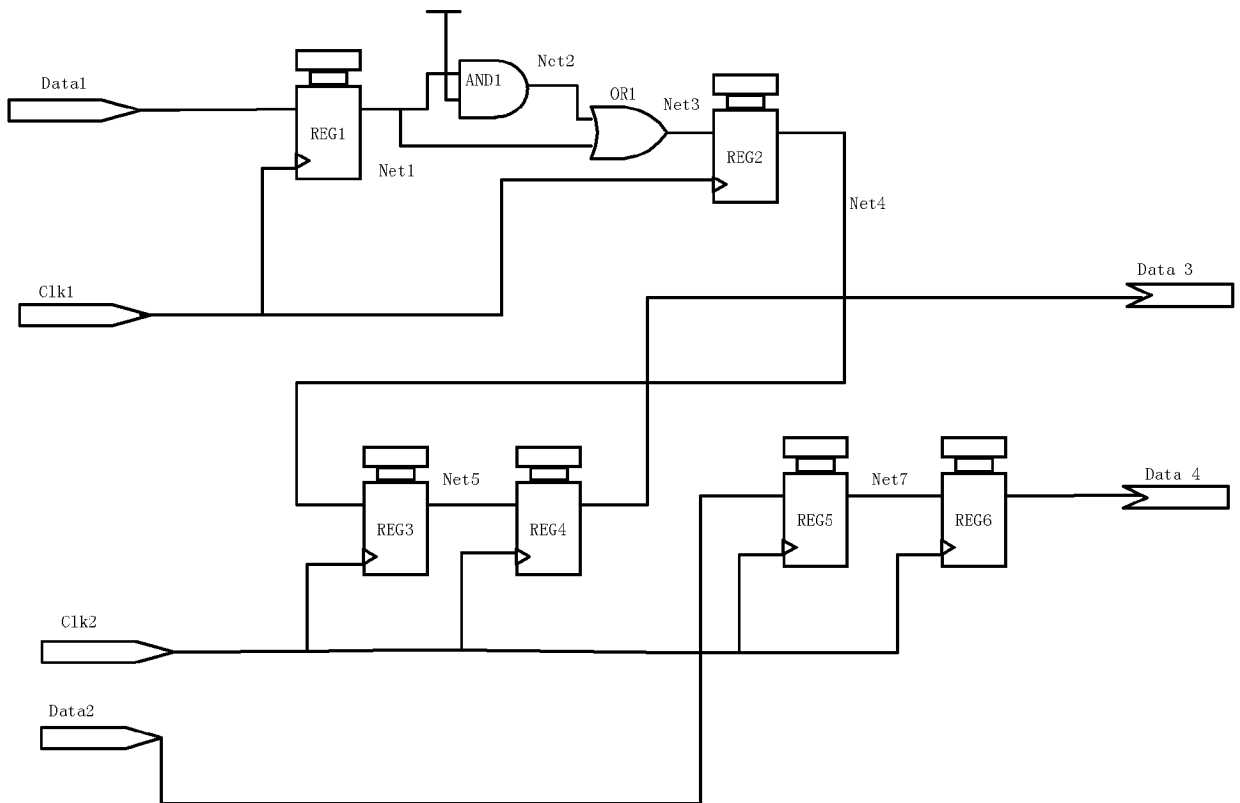


图 4

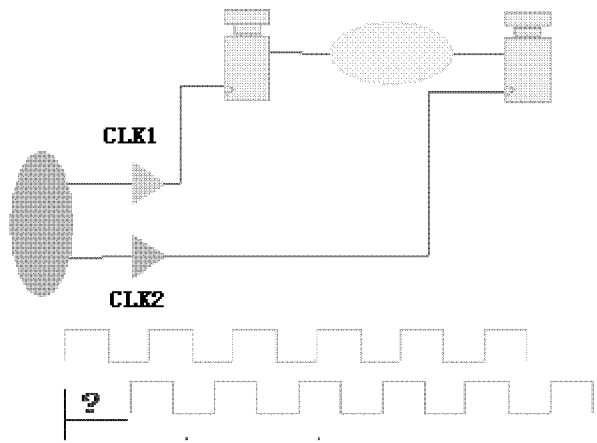


图 5

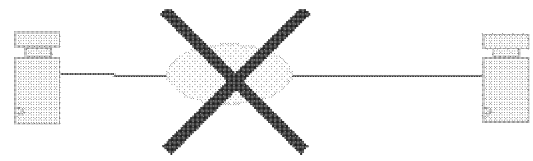


图 6

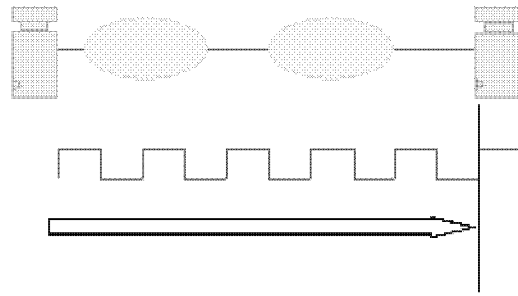


图 7

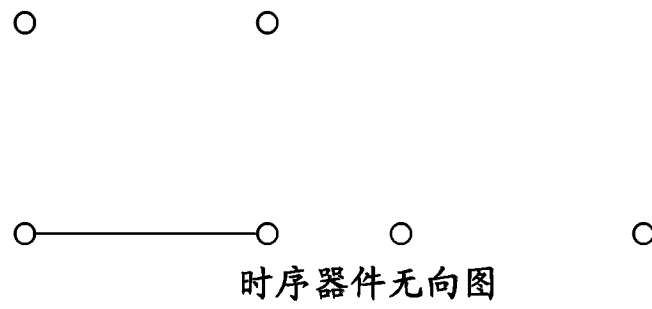
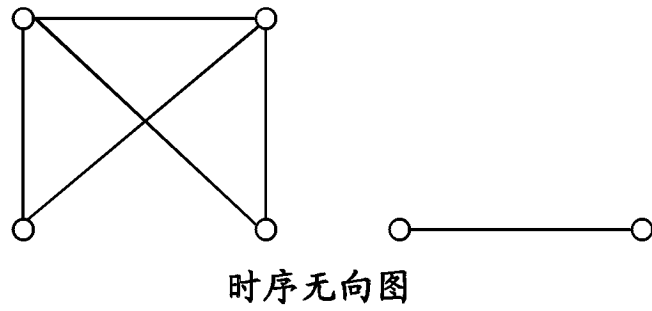
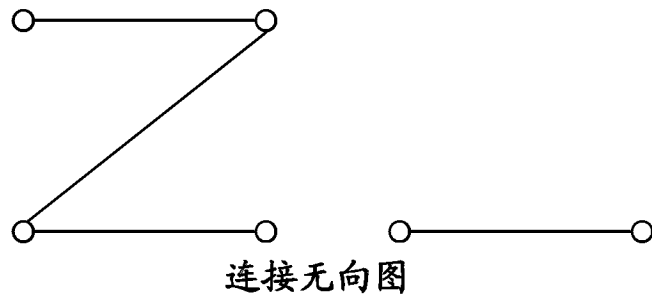
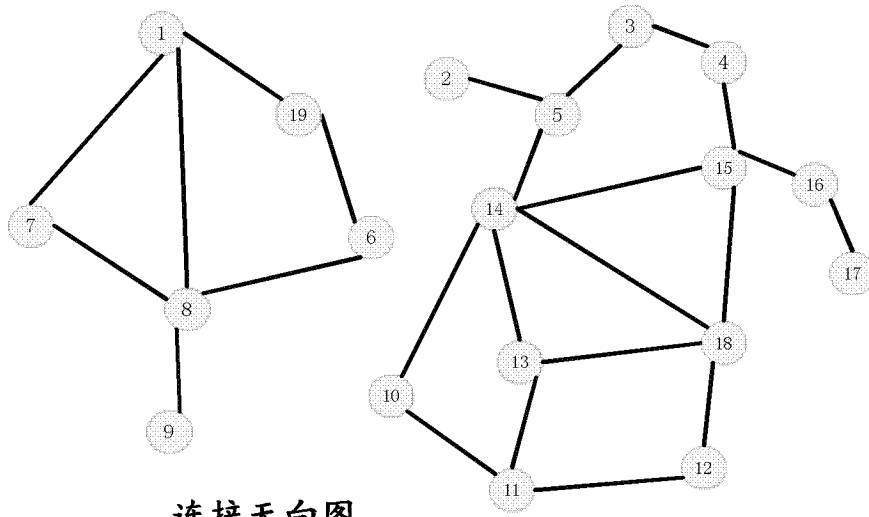
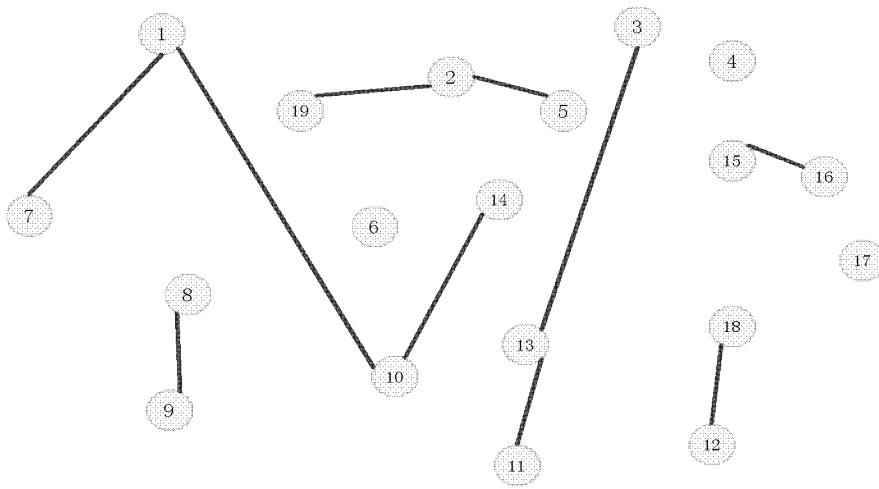


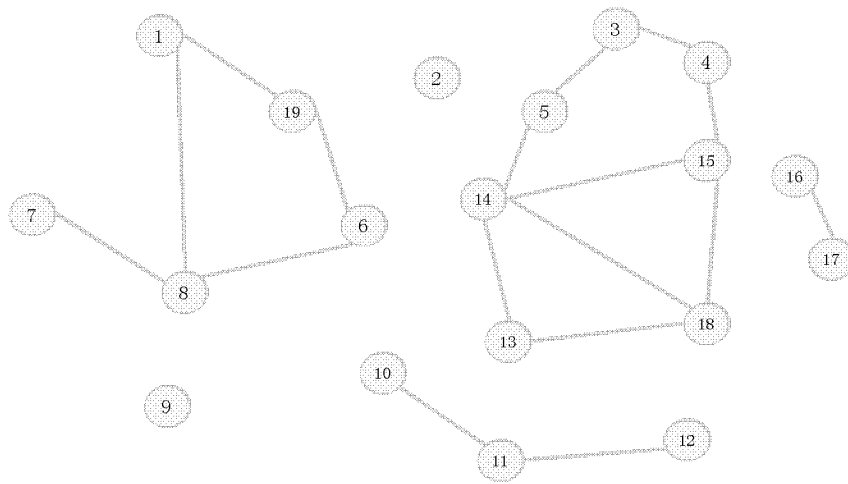
图 8



连接无向图



时序无向图



时序器件无向图

图 9

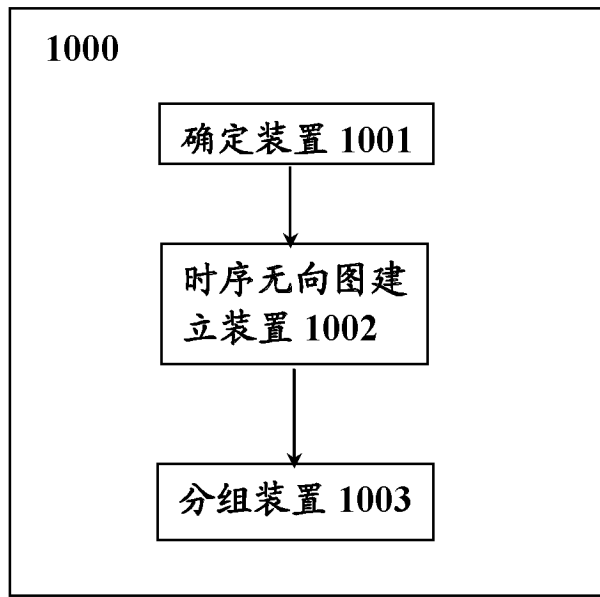


图 10