



(12)发明专利

(10)授权公告号 CN 105205205 B

(45)授权公告日 2018.08.28

(21)申请号 201510507681.5

(22)申请日 2015.08.18

(65)同一申请的已公布的文献号  
申请公布号 CN 105205205 A

(43)申请公布日 2015.12.30

(73)专利权人 北京大学  
地址 100871 北京市海淀区颐和园路5号

(72)发明人 罗国杰 沈明华

(74)专利代理机构 北京万象新悦知识产权代理  
事务所(普通合伙) 11360  
代理人 张肖琪

(51)Int.Cl.  
G06F 17/50(2006.01)

(56)对比文件

US 5818730 A,1998.10.06,  
US 7310594 B1,2007.12.18,  
US 2011055784 A1,2011.03.03,  
CN 103246554 A,2013.08.14,  
CN 103886137 A,2014.06.25,  
陈迅.《面向FPGA设计及应用的EDA关键技术  
研究》.《中国博士学位论文全文数据库 信息科  
技辑》.2014,(第4期),

审查员 张念

权利要求书2页 说明书8页 附图8页

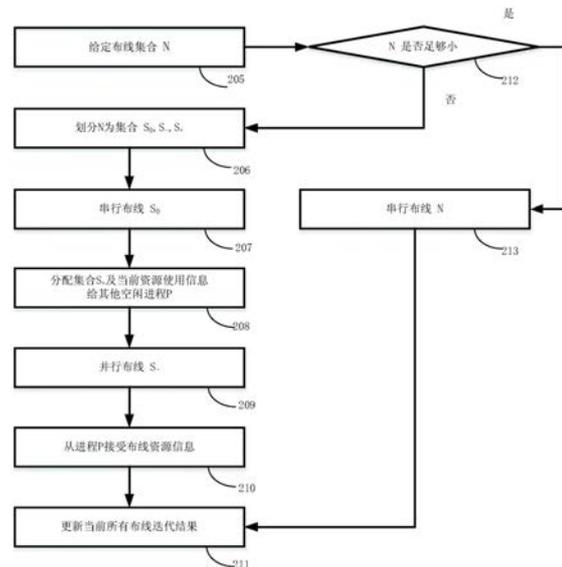
(54)发明名称

基于网表位置信息最优划分的FPGA粗粒度  
并行布线方法

(57)摘要

本发明公布了一种基于网表位置信息最优划分的FPGA粗粒度并行布线方法,通过递归方法进行任务划分并同步任务间的布线信息,完成基于当前惩罚代价搜索所有信号线的布线路径;针对布线集合N,若集合足够小则采用串行布线方法;若不满足足够小,将N划分为集合S<sub>-</sub>、S<sub>0</sub>和S<sub>+</sub>;分别是横跨两个子区域的信号线集合S<sub>0</sub>、包含仅在左方(或下方)子区域的信号线集合S<sub>-</sub>和包含仅在右方(或上方)子区域的信号线集合S<sub>+</sub>;通过递归划分方法继续对S<sub>-</sub>和S<sub>+</sub>沿着不同方向再次进行任务划分,产生二叉树形式的信号线集合;进行并行布线操作,完成布线过程。本发明能够提高FPGA的开发效率,节约工作时间成本,促进FPGA广泛使用。

CN 105205205 B



1. 一种基于网表位置信息最优划分的FPGA粗粒度并行布线方法,通过递归方法进行任务划分,使用多进程方法执行任务并同步任务间的布线信息,完成基于当前惩罚代价搜索所有信号线的布线路径;包括如下步骤:

1) 针对布线集合N,若集合N足够小则采用串行布线方法;若N不满足足够小条件,则将布线集合N划分为集合 $S_-$ 、 $S_0$ 和 $S_+$ ;将布线区域划分为左方和右方两个子区域,或者划分为上方和下方两个子区域; $S_0$ 为横跨两个子区域的信号线集合;当布线区域被划分为左方和右方两个子区域时, $S_-$ 与 $S_+$ 分别为包含仅在左方子区域的信号线集合与包含仅在右方子区域的信号线集合;当布线区域被划分为上方和下方两个子区域时, $S_-$ 与 $S_+$ 分别为包含仅在下方子区域的信号线集合与包含仅在上方子区域的信号线集合;若N划分后的三个集合 $S_-$ 、 $S_0$ 和 $S_+$ 拥有相似数目的信号线,则集合N为足够小;

2) 通过递归划分方法继续对集合 $S_-$ 和 $S_+$ 沿着不同方向采用步骤1)的方法再次进行任务划分,通过递归划分过程产生二叉树形式的信号线集合;

3) 对步骤2)得到的二叉树形式的信号线集合进行并行布线操作,完成布线过程。

2. 如权利要求1所述基于网表位置信息最优划分的FPGA粗粒度并行布线方法,其特征是,步骤3)通过多进程方法执行并行布线操作,具体包括如下步骤:

31) 主进程利用信号线划分获得三类信号线集合,完成自己负责的横跨区域信号线集合内的所有布线任务后,把当前的部分布线结果和区域内信号线集合的布线任务分配给相应的进程;

32) 收到布线任务的进程通过相同的方法进行信号线划分、串行布线和并行任务分配,形成递归的过程;

33) 任意进程完成各自的布线任务后,将布线结果返回父级进程;父级进程完成布线结果合并后,通过同样方法将布线结果返回祖父级进程,以此类推;

34) 最后,主进程获得所有的布线结果,并判断是否存有布线资源的拥塞;若没有布线资源的拥塞,则布线完成;若有布线资源的拥塞,则更新布线资源节点的代价,并且回到步骤31)重新布线,直至所有拥塞消除。

3. 如权利要求1所述基于网表位置信息最优划分的FPGA粗粒度并行布线方法,其特征是,步骤2)所述任务划分通过采用多项式时间内获得最优划分的方法来更好地加速并行布线,具体包括如下步骤:

21) 进行预处理:假设布通一条信号线 $n$ 的时间为 $t(n)$ ,对任意区域 $R$ ,预计算出包围盒恰为 $R$ 的所有信号线的串行布线时间,记为 $t(\sigma(R))$ ;假设区域 $R$ 被分割线 $p$ 划分成 $R_{p,-}$ 和 $R_{p,+}$ ,区域 $R$ 内的信号线集合 $S(R)$ 被划分成三个集合 $S_0\langle R,p\rangle$ 、 $S(R_{p,-})$ 和 $S(R_{p,+})$ ;预计算出 $S_0\langle R,p\rangle$ 的串行布线时间,记为 $t(S_0\langle R,p\rangle)$ ;继而通过式1预计算出 $S(R)$ 的串行布线时间 $t(S(R))$ :

$$t(S(R)) = t(S_0\langle R,p\rangle) + t(S(R_{p,-})) + t(S(R_{p,+})) \quad (\text{式1})$$

其中, $t(S(R_{p,-}))$ 和 $t(S(R_{p,+}))$ 分别为 $S(R_{p,-})$ 和 $S(R_{p,+})$ 的串行布线时间;

22) 根据步骤21)的预计算结果,执行动态规划处理:给定划分/递归的最大层数 $L$ ,定义 $T(S(R),L)$ 为对布线区域 $R$ 使用递归并行布线方法的估算布线时间;

初始状态: $T(S(R),0) = t(S(R))$ ;  $t(S(R))$ 为 $S(R)$ 的串行布线时间;

状态转移方程:

$$T(S(R), L + 1) = \min \left\{ \begin{array}{c} T(S(R), L) \\ \min_p \{ t(S_0(R, p)) + \max\{T(S(R_{p,-}), L), T(S(R_{p,+}), L)\} \} \end{array} \right\} \quad (\text{式 2})$$

23) 根据步骤22) 得到的估计布线时间来得到信号线划分。

4. 如权利要求3所述基于网表位置信息最优划分的FPGA粗粒度并行布线方法,其特征是,步骤22) 所述划分/递归的最大层数L为具体电路的最优递归层数,通过依次增加递归层数来估计布线时间,直至估计的布线时间不再缩短,从而确定具体电路的最优递归层数。

## 基于网表位置信息最优划分的FPGA粗粒度并行布线方法

### 技术领域

[0001] 本发明属于电子设计自动化领域,涉及电子设计自动化布线方法,尤其涉及一种基于网表位置信息最优划分的FPGA粗粒度并行布线方法。

### 背景技术

[0002] 随着现场可编程门阵列(field programmable gate array,FPGA)规模的持续大幅度递增,与其相对应的电子设计自动化(electrical design automatic,EDA)工具需要花费大量运行时间来综合处理这些大规模的电路设计。目前,有希望从两个角度来缓解这个问题:一是利用工程设计方法学的模块化、层次化和设计重用;二是依靠多核和众核计算平台,通过并行化来加速EDA工具从而减轻运行时间过长所带来的挑战。

[0003] 布线是整套EDA设计流程中占用时间最长的步骤之一。布线的整体流程如图1所示,首先进行布局设计;然后计算布线资源节点的历史惩罚代价;再基于当前惩罚代价搜索得到所有信号线的布线路径;判断上述布线路径是否存在拥塞;若存在拥塞则循环执行计算布线资源节点的历史惩罚代价,再基于当前惩罚代价搜索得到所有信号线的布线路径,并判断上述布线路径是否存在拥塞;若不存在拥塞则进行布线设计。针对基于当前惩罚代价搜索得到所有信号线的布线路径的过程,目前最流行的布线器是采用基于拥塞协商的路径探索算法,它的基本工作原理就是不断循环迭代的过程,如图2.1所示,从给定布线集合取出一条信号线,布通该信号线,循环布通下一条信号线,直到布通所有信号;最后更新当前迭代结果得到所有信号线的布线路径。因此,现有方法具体是在初始布线环节,算法暂时允许存在拥塞的情况下,布通所有的信号线;紧接着对产生拥塞的信号线施加惩罚代价,然后根据新代价重新布通所有的(或拥塞的)信号线,最后判断检查是否仍存在有拥塞的信号线,如果没有,那么布线工作完成;如果依旧存有,那么继续更新代价并进入重新布线阶段,直到没有拥塞产生,最终完成布线工作。

[0004] 现有的这种方法依靠断开与重新连接的串行布线工作方式非常浪费时间,而且随着FPGA密度的增大,将会占用更多的时间,从而影响工程生产力,提高工作成本,并对软件开发者广泛地使用FPGA产生一定的阻碍。

### 发明内容

[0005] 为了克服上述现有技术的不足,本发明提供一种基于网表位置信息最优划分的FPGA粗粒度并行布线方法,该方法是在现有多核处理器环境中面向加速FPGA布线的并行化设计方法,通过递归的方式进行任务划分和同步任务间的布线信息,并且在此方式下可采用最优的任务划分策略,通过并行化布线来加速FPGA布线。

[0006] 本发明的原理是:FPGA布线的整体流程包括布局设计;计算布线资源节点的历史惩罚代价;基于当前惩罚代价搜索所有信号线的布线路径;判断布线路径是否存在拥塞;不存在拥塞则进行布线设计。在布线整套流程中,针对基于当前惩罚代价搜索得到所有信号线的布线路径的过程,考虑到现有的计算环境通常是多核平台,可以利用多核处理器的计

算能力对此算法进行并行处理,从而加快布线工作,缩短整个电路综合流程的时间。网表位置信息为网表的每个信号线的源点和汇点对应的布线资源节点的物理坐标;并行布线方法分为粗粒度和细粒度两类,其中细粒度方法对单一信号线搜索布线路径的任务进行划分和并行处理,而粗粒度方法对整体多条信号线的布线任务进行分配和并行处理。在布线整套流程中,针对基于当前惩罚代价搜索得到所有信号线的布线路径的过程,相比现有方法采用的基于拥塞协商的路径探索方法,本发明提出了一种基于网表位置信息最优划分的FPGA粗粒度并行布线方法(ParRoute),该方法是在现有多核处理器环境中提供一种面向加速FPGA布线的并行化设计方法,通过递归的方式进行任务划分以及同步任务间的布线信息,将布线集合N划分为集合S<sub>-</sub>、S<sub>0</sub>和S<sub>+</sub>;分别是横跨两个子区域的信号线集合S<sub>0</sub>、包含仅在左方(或下方)子区域的信号线集合S<sub>-</sub>和包含仅在右方(或上方)子区域的信号线集合S<sub>+</sub>;并且提出在此方式下最优的任务划分策略,完成基于当前惩罚代价搜索所有信号线的布线路径。本发明进一步提出增强的FPGA粗粒度并行布线方法(ParRoute+),运用同样方法来递归划分以及并行化处理 ParRoute的串行任务(即针对横跨两个子区域的信号线集合S<sub>0</sub>)。

[0007] 本发明提供的技术方案是:

[0008] 一种基于网表位置信息最优划分的FPGA粗粒度并行布线方法,通过递归方法进行任务划分,使用多进程方法执行任务并同步任务间的布线信息,完成基于当前惩罚代价搜索所有信号线的布线路径;包括如下步骤:

[0009] 1) 针对布线集合N,若集合N足够小则采用串行布线方法;若N不满足足够小的条件,则将布线集合N划分为集合S<sub>-</sub>、S<sub>0</sub>和S<sub>+</sub>;分别是横跨两个子区域的信号线集合S<sub>0</sub>、包含仅在左方(或下方)子区域的信号线集合S<sub>-</sub>和包含仅在右方(或上方)子区域的信号线集合S<sub>+</sub>;

[0010] 2) 通过递归划分方法继续对集合S<sub>-</sub>和S<sub>+</sub>沿着不同方向采用步骤1)的方法再次进行任务划分,通过递归划分过程产生二叉树形式的信号线集合;

[0011] 3) 对步骤2)得到的二叉树形式的信号线集合进行并行布线操作,完成布线过程。

[0012] 针对上述基于网表位置信息最优划分的FPGA粗粒度并行布线方法,进一步地,步骤3) 使用多进程方法执行任务并同步任务间的布线信息,即通过多进程方法实现所述任务划分和并行布线操作,具体包括如下步骤:

[0013] 31) 主进程利用信号线划分获得三类信号线集合,完成自己负责的横跨区域信号线集合内的所有布线任务后,把当前的部分布线结果和区域内信号线集合的布线任务分配给相应的进程;

[0014] 32) 收到布线任务的进程通过相同的方法进行信号线划分、串行布线和并行任务分配,形成递归的过程;

[0015] 33) 任意进程完成各自的布线任务后,将布线结果返回父级进程;父级进程完成布线结果合并后,通过同样方法将布线结果返回祖父级进程,以此类推;

[0016] 34) 最后,主进程获得所有的布线结果,并判断是否存有布线资源的拥塞;若没有布线资源的拥塞,则布线完成;若有布线资源的拥塞,则更新布线资源节点的代价,并且回到步骤31)重新布线,直至所有拥塞消除。

[0017] 在本发明实施例中,针对五种电路blob\_merge、or1200、mcml、LU8EEng和mkDelayW,上述多进程方法采用的进程数目分别为1,2,4,8,16和32个。

[0018] 针对上述基于网表位置信息最优划分的FPGA粗粒度并行布线方法,进一步地,步

骤2) 所述任务划分通过采用多项式时间内获得最优划分的方法来更好地加速并行布线, 具体包括如下步骤:

[0019] 21) 进行预处理: 假设布通一条信号线 $n$ 的时间为 $t(n)$ , 对任意区域 $R$ , 预计算出包围盒恰为 $R$ 的所有信号线的串行布线时间, 记为 $t(\sigma(R))$ ; 假设区域 $R$ 被分割线 $p$ 划分成 $R_{p,-}$ 和 $R_{p,+}$ , 区域 $R$ 内的信号线集合 $S(R)$ 被划分成三个集合 $S_0\langle R, p \rangle$ 、 $S(R_{p,-})$ 和 $S(R_{p,+})$ ; 预计算出 $S_0\langle R, p \rangle$ 的串行布线时间, 记为 $t(S_0\langle R, p \rangle)$ ; 继而通过式1预计算出 $S(R)$ 的串行布线时间 $t(S(R))$ :

$$[0020] \quad t(S(R)) = t(S_0\langle R, p \rangle) + t(S(R_{p,-})) + t(S(R_{p,+})) \quad (\text{式1})$$

[0021] 其中,  $t(S(R_{p,-}))$ 和 $t(S(R_{p,+}))$ 分别为 $S(R_{p,-})$ 和 $S(R_{p,+})$ 的串行布线时间, 可通过合适的计算顺序得到, 成为已知的预计算结果; 具体地, 对于不可再划分的最小区域的串行布线时间可直接求出, 而其他区域的串行布线时间按区域面积从小到大的顺序用式1计算; 当求取式1左边项时, 右边各项的区域面积比左边项的区域小, 计算顺序可确保右边各项的值为已知;

[0022] 22) 根据步骤21)的预计算结果, 执行动态规划处理:

[0023] 给定划分/递归的最大层数 $L$ , 定义 $T(S(R), L)$ 为对布线区域 $R$ 使用递归并行布线方法的估算布线时间, 具体过程为:

[0024] 初始状态:  $T(S(R), 0) = t(S(R))$ ;  $t(S(R))$ 为 $S(R)$ 的串行布线时间;

[0025] 状态转移方程:

$$[0026] \quad T(S(R), L+1) = \min \left\{ \begin{array}{l} T(S(R), L) \\ \min_p \{ t(S_0\langle R, p \rangle) + \max \{ T(S(R_{p,-}), L), T(S(R_{p,+}), L) \} \} \end{array} \right\} \quad (\text{式2})$$

[0027] 式2中, 设定区域 $R$ 被分割线 $p$ 划分成 $R_{p,-}$ 和 $R_{p,+}$ , 区域 $R$ 内的信号线集合 $S(R)$ 被划分成三个集合 $S_0\langle R, p \rangle$ 、 $S(R_{p,-})$ 和 $S(R_{p,+})$ ;  $t(S_0\langle R, p \rangle)$ 为预计算出 $S_0\langle R, p \rangle$ 的串行布线时间;

[0028] 23) 根据步骤22)得到的估计布线时间来获得信号线划分。

[0029] 根据得到的估计布线时间来确定信号线划分, 由上述方法确定的信号线划分可获得全局最优解, 即若布通一条信号线 $n$ 的估算时间为 $t(n)$ 且式1的估算方法准确, 则用此方法确保按所求位置进行划分并行的时间是最短的, 从而更好地加速并行布线。该方法对于每一个进程中串行处理的横跨型集合的信号线, 可将横跨型集合的信号线也划分成三类集合, 均使用相同的递归并行方法处理, 从而获得更大的并行度。

[0030] 上述通过采用多项式时间内获得最优划分的方法来划分任务, 从而更好地加速并行布线, 在本发明实施例中, 所述递归的最大层数为3。对于一般情况, 可依次增加递归层数 $L$ 来估计布线时间, 直至估计的布线时间不再缩短, 从而确定具体电路的最优递归层数。

[0031] 上述基于网表位置信息最优划分的FPGA粗粒度并行布线方法, 进一步地, 步骤1)中,  $N$ 划分后三个集合 $S - \cup S_0 \cup S_+$ , 若 $S_-$ 、 $S_0$ 和 $S_+$ 拥有相似数目的信号线, 则认为集合 $N$ 足够小。集合 $N$ 足够小是给划分以足够的保证, 进而使得并行更加有效。

[0032] 采用上述基于网表位置信息最优划分的FPGA粗粒度并行布线方法, 通过并行化布线来加速FPGA布线, 可获得至少7倍的加速。

[0033] 与现有技术相比, 本发明的有益效果是:

[0034] 在FPGA开发中, 现有方法依靠断开与重新连接进行串行布线, 需要时间长, 而且随着FPGA密度的增大, 将会占用更多的时间, 从而影响工程生产力, 工作成本高。而本发明提

供一种基于网表位置信息最优划分的FPGA粗粒度并行布线方法,该方法是在现有多核处理器环境中面向加速FPGA布线的并行化设计方法,通过递归的方式进行任务划分和同步任务间的布线信息,并且在此方式下可采用最优的任务划分策略,通过并行化布线来加速FPGA布线。使用本发明能够获得至少7倍的加速,进而提高FPGA的开发效率,节约工作时间成本,并促进FPGA更广泛的使用。

## 附图说明

[0035] 图1是FPGA布线的整体流程框图;

[0036] 其中,100—布局设计;101—计算布线资源节点的历史惩罚代价;102—基于当前惩罚代价搜索所有信号线的布线路径;103—判断是否存在拥塞;104—布线设计。

[0037] 图2是基于当前惩罚代价搜索得到所有信号线的布线路径的过程步骤现有方法和本发明方法的流程框图;其中,2.1是现有方法的实现流程框图;2.2是本发明提供方法的实现流程框图;200—给定布线集合;201—从布线集合中取出一条信号线;202—布通此信号线;203—判断是否是最后一条信号线;204—更新当前迭代结果;205—给定布线集合;206—将给定集合划分成三类集合 $S_0, S_-, S_+$ ;207—以串行布线方式布通 $S_0$ ;208—分配集合 $S_+$ 及当前资源使用信息给其他空闲进程;209—并行布线 $S_-$ ;210—从空闲进程接受布线资源信息;211—更新当前所有布线迭代结果;212—判断给定布线集合是否足够小;213—如果足够小就执行串行布线。

[0038] 图3是本发明实施例中经过第一次划分产生横跨两个子区域的信号线集合 $S_0$ 、包含仅在左方(或下方)子区域的信号线集合 $S_-$ 和包含仅在右方(或上方)子区域的信号线集合 $S_+$ 的示意图;

[0039] 其中,301为第一次划分; $S_-$ 包含信号线300; $S_0$ 包含信号线302、303和304; $S_+$ 包含信号线305;306圆点表示信号线的源点,307叉点表示信号线的汇点。

[0040] 图4是本发明实施例中对横跨两个子区域的信号线集合 $S_0$ 进行一次划分产生三类集合 $S_{0-}, S_{00}, S_{0+}$ 的示意图;

[0041] 其中,404为对 $S_0$ 进行一次划分; $S_{0-}$ 包含信号线401; $S_{00}$ 包含信号线402; $S_{0+}$ 包含信号线403。

[0042] 图5是本发明实施例中经过三次划分的情形示意图;

[0043] 其中,500为第一次划分;501和502为第二次划分;503、504、505和506为第三次划分。

[0044] 图6是本发明实施例中经过三次划分后产生的信号线集合分配给八个进程的过程示意图;

[0045] 其中,进程600依次布通 $S_0, S_-, S_{--}$ 和 $S_{---}$ 中的信号线;其他进程的布线方式与进程600一致。

[0046] 图7是本发明实施例中对 $S_0$ 作进一步并行处理;

[0047] 其中,对 $S_0$ 进行一次划分404,会产生三类集合的并行方式。进程700布通集合 $S_{0-}$ 和 $S_{00}$ 中的信号线。进程701布通集合 $S_{0+}$ 中的信号线。

[0048] 图8是本发明实施例中通过并行处理 $S_0$ 后带来运行时间缩减的示意图;

[0049] 其中,并行化801由 $S_0$ 进行一次划分404产生的三类集合,所带来运行时间的减少

为805；802,803,804,806和807分别表示进程间的同步。

[0050] 图9是本发明实施例中五种电路在不同进程数目的情况下迭代次数的变化情况的示意图；

[0051] 其中,901—电路blob\_merge;902—电路or1200;903—电路mcml;904—电路LU8EEng;905—电路mkDelayW。

[0052] 图10是本发明实施例中905电路mkDelayW在并行布线过程中,随着迭代次数的增加,拥塞节点数目的变化情况的示意图；

[0053] 其中,1001表示集合 $S_0$ 和 $S_-$ 中拥塞节点的数目变化;1002表示集合 $S_0$ 和 $S_+$ 中拥塞节点的数目变化;1003表示集合 $S_0$ 中拥塞节点的变化。

[0054] 图11为本发明实施例中905电路mkDelayW在并行布线过程中,随着迭代次数的增加,拥塞节点数目的变化情况的示意图；

[0055] 其中,2001表示集合 $S_-$ 中拥塞节点的数目变化;2002表示集合 $S_+$ 中拥塞节点的数目变化;2003表示集合 $S_0$ 中拥塞节点的变化。

### 具体实施方式

[0056] 下面结合附图,通过实施例进一步描述本发明,但不以任何方式限制本发明的范围。

[0057] 由FPGA芯片内布线资源抽象出来的布线资源图中,每一条需要布通的信号线都具有一个源点和一个或多个汇点;通过使用布线资源图中的布线资源节点连通源点和汇点,则获得一棵布线树;当获得所有信号线的布线树、且不同信号的布线树没有共用同一个布线资源节点,方可完成所有的布线工作。由于此类信号线本身的源点与汇点在布线资源图中的坐标位置已知,因此,本技术方案可以通过节点坐标与分割线的位置关系对信号线进行分类。

[0058] 假设分割线是布线资源图的中位线,本技术方案对信号线进行划分时,利用此分割线将布线区域划分成两个子区域从而得到三类信号线的集合,分别是横跨两个子区域的信号线集合 $S_0$ 、包含仅在左方(或下方)子区域的信号线集合 $S_-$ 、以及包含仅在右方(或上方)子区域的信号线集合 $S_+$ 。

[0059] 图1是FPGA布线的整体流程框图;包括布局设计;计算布线资源节点的历史惩罚代价;基于当前惩罚代价搜索所有信号线的布线路径;判断布线路径是否存在拥塞;不存在拥塞则进行布线设计。在布线整套流程中,针对基于当前惩罚代价搜索得到所有信号线的布线路径的过程,相比现有方法采用的基于拥塞协商的路径探索方法,本发明提出了一种基于网表位置信息最优划分的FPGA粗粒度并行布线方法(ParRoute),通过递归方法进行任务划分并同步任务间的布线信息,完成基于当前惩罚代价搜索所有信号线的布线路径;图2是基于当前惩罚代价搜索得到所有信号线的布线路径的过程步骤现有方法和本发明方法的流程框图;其中,2.1是现有方法的实现流程框图;2.2是本发明提供方法的实现流程框图;图2.1是现有方法的实现流程框图,具体是逐条从布线集合中取出信号线进行布线,最后更新得到布线结果,现有布线方法为串行布线方法。2.2是本发明提供方法的实现流程框图;主要包括如下步骤:

[0060] 1) 针对布线集合 $N$ ,若集合 $N$ 足够小则采用串行布线方法;若 $N$ 不满足足够小的条

件,则将布线集合N划分为集合S<sub>-</sub>、S<sub>0</sub>和S<sub>+</sub>;分别是横跨两个子区域的信号线集合S<sub>0</sub>、包含仅在左方(或下方)子区域的信号线集合S<sub>-</sub>和及包含仅在右方(或上方)子区域的信号线集合S<sub>+</sub>;

[0061] 2)通过递归划分方法继续对集合S<sub>-</sub>和S<sub>+</sub>沿着不同方向采用步骤1)的方法再次进行任务划分,通过递归划分过程产生二叉树形式的信号线集合;

[0062] 3)对步骤2)得到的二叉树形式的信号线集合进行并行布线操作,完成布线过程。

[0063] 针对布线集合N,本实验中规定若N划分后的三个集合S<sub>-</sub>∪S<sub>0</sub>∪S<sub>+</sub>拥有相似数目的信号线,则集合N足够小,采用串行布线方法;若N不满足足够小,则划分N为集合S<sub>-</sub>,S<sub>0</sub>,S<sub>+</sub>;以串行布线方式布通S<sub>0</sub>;分配集合S<sub>+</sub>及当前资源使用信息给其他空闲进程;并行布线S<sub>-</sub>;从空闲进程接受布线资源信息;更新当前所有布线迭代结果;其执行操作如图2.2所示。本发明方法(ParRoute)使用图2.2的流程实现图1中的步骤102,在图2.2中,步骤207和步骤213使用图2.1中的步骤实现,步骤209使用图2.2实现递归划分布线。

[0064] 本发明提出的基于网表位置信息最优划分的FPGA粗粒度并行布线方法(ParRoute)通过递归并行布线,具体是对信号线进行划分为上述三类信号线的集合后,继续对集合S<sub>-</sub>和S<sub>+</sub>沿着不同方向使用同样的方法再次进行划分操作,进而形成一个递归划分的过程。图3的从粗到细的实线显示了三层划分的过程。集合S<sub>-</sub>(或S<sub>+</sub>)被划分成中集合S<sub>-0</sub>、S<sub>--</sub>与S<sub>++</sub>(或S<sub>0</sub>、S<sub>--</sub>与S<sub>++</sub>)。S<sub>-0</sub>与S<sub>0</sub>留在当前层级,而S<sub>--</sub>、S<sub>-1</sub>、S<sub>+-</sub>和S<sub>++</sub>会继续被划分处理。基于以上分类方式产生信号线集合的过程形成二叉树。对此二叉树的并行操作描述如下(假设使用多进程实现),该过程包括如下步骤:

[0065] 31)主进程利用信号线划分获得三类集合,完成自己负责的横跨区域信号线集合内的所有布线任务后,把当前的部分布线结果和区域内信号线集合的布线任务分配给相应的进程。

[0066] 32)收到布线任务的进程,使用同样的模式进行信号线划分、串行布线、以及并行任务分配,形成递归的过程。

[0067] 33)任意进程完成各自的布线任务后,将布线结果返回父级进程;父级进程完成布线结果合并后,以同样方式将布线结果返回祖父级进程,以此类推。

[0068] 34)最后主进程获得所有的布线结果,并判断是否存有布线资源的拥塞。若没有,布线完成;若有,更新布线资源节点的代价,并且回到步骤31)重新布线,直至所有拥塞消除。

[0069] 在上述基于网表位置信息最优划分的FPGA粗粒度并行布线方法(ParRoute)的基础上,本发明还提供一种最优划分方法。最优的划分方式需满足每一层中两个子区域布线任务的均衡性。本技术方案提供一种多项式时间内获得最优划分的算法来更好地加速并行布线,步骤2)的具体优化方法如下:

[0070] 21)预处理:假设布通一条信号线n的时间为t(n),对任意区域R,可预计算出包围盒恰为R的所有信号线的串行布线时间,记为t(σ(R));假设区域R被分割线p划分成R<sub>p,-</sub>和R<sub>p,+</sub>,区域R内的信号线集合S(R)被划分成三个集合S<sub>0</sub><R,p>、S(R<sub>p,-</sub>)和S(R<sub>p,+</sub>);可预计算出S<sub>0</sub><R,p>的串行布线时间,记为t(S<sub>0</sub><R,p>);继而可预计算出S(R)的串行布线时间t(S(R)):

[0071] 
$$t(S(R)) = t(S_0\langle R, p \rangle) + t(S(R_{p,-})) + t(S(R_{p,+})) \quad (\text{式1})$$

[0072] 其中,t(S(R<sub>p,-</sub>))和t(S(R<sub>p,+</sub>))可由合适的计算顺序成为已知的预计算结果;

[0073] 22) 动态规划:根据以上的预计算结果,开始执行动态规划处理:给定划分/递归的最大层数L(可依次增加递归层数L来估计布线时间,直至估计的布线时间不再缩短,从而确定具体电路的最优递归层数;针对本发明实施例,图6中L为3),定义 $T(S(R),L)$ 为对布线区域R使用递归并行布线方法的估算布线时间;

[0074] 初始状态: $T(S(R),0) = t(S(R))$ ;

[0075] 状态转移方程:

$$[0076] \quad T(S(R),L+1) = \min \left\{ \begin{array}{l} T(S(R),L) \\ \min_p \{ t(S_0(R,p)) + \max \{ T(S(R_{p,-}),L), T(S(R_{p,+}),L) \} \} \end{array} \right\} \quad (\text{式2})$$

[0077] 由上述方法确定的信号线划分可获得全局最优解,即若布通一条信号线n的估算时间 $t(n)$ 以及式1的估算方法准确,则用此方法确保按所求位置进行划分并行的时间是最短的,从而更好地加速并行布线。该方法对于每一个进程中串行处理的横跨型集合的信号线,可将横跨型集合的信号线也划分成三类集合,均使用相同的递归并行方法处理,从而获得更大的并行度。再次对 $S_0$ 进行并行布线的过程称为ParRouter+。

[0078] 以下通过实施例说明本发明技术方案:

[0079] 如图3所示,信号线的源点由306表示,而汇点由307表示,经过第一次划分301,会产生三类集合,分别是横跨两个子区域的信号线集合 $S_0$ 、包含仅在左方(或下方)子区域的信号线集合 $S_-$ 、以及包含仅在右方(或上方)子区域的信号线集合 $S_+$ ,其中 $S_-$ 包含信号线300, $S_0$ 包含信号线302、303和304, $S_+$ 包含信号线305。如图4所示,对 $S_0$ 进行一次划分404,会产生三类集合,分别是 $S_{0-}$ 、 $S_{00}$ 、 $S_{0+}$ ,其中 $S_{0-}$ 包含信号线401, $S_{00}$ 包含信号线402, $S_{0+}$ 包含信号线403。

[0080] 如图5所示,经过三次划分的情形,第一次划分500,第二次划分501和502,第三次划分503、504、505和506。图6是经过三次划分后产生的信号线集合分配给八个进程的过程。进程600依次布通 $S_0$ 、 $S_-$ 、 $S_{--}$ 、 $S_{---}$ 中的信号线。其他进程的布线方式与进程600一致。

[0081] 如图7所示,对 $S_0$ 进行一次划分404,会产生三类集合的并行方式。进程600布通集合 $S_{0-}$ 和 $S_{00}$ 中的信号线。进程601布通集合 $S_{0+}$ 中的信号线。如图8所示,并行化801由 $S_0$ 进行一次划分404产生的三类集合,所带来运行时间的减少805。其中802、803、804、806、807分别表示进程间的同步。

[0082] 本技术方案对本发明提供方法的布线收敛性进行了测试分析,测试了具有代表性的大规模FPGA电路,依次增加进程的数目来观察迭代次数的变化。如图9所示,五种电路在不同进程数目的情况下(五种电路是blob\_merge、or1200、mcml、LU8EEng和mkDelayW.,分别在进程数目为1,2,4,8,16,32情况下),迭代次数的变化情况;随着进程数目的递增,迭代次数会有减少的趋势,这是因为本技术方案的并行布线器ParRouter/ParRouter+显式地控制了不同信号线的布线顺序,从而加快了布线资源拥塞度的下降过程。

[0083] 本技术方案同时测试每次迭代过程中,拥塞节点的分布情况。如图10所示,为电路905在并行布线过程中,随着迭代次数的增加,拥塞节点数目的变化情况,其中1001表示集合 $S_0$ 和 $S_-$ 中拥塞节点的数目变化,1002表示集合 $S_0$ 和 $S_+$ 中拥塞节点的数目变化,1003表示集合 $S_0$ 中拥塞节点的变化。图10展示了信号线集合 $S_0 \cup S_-$ 和 $S_0 \cup S_+$ 的拥塞节点数目、以及它们共同的拥塞节点数目的下降趋势。

[0084] 如图11所示,为电路905在并行布线过程中,随着迭代次数的增加,拥塞节点数目

的变化情况,其中2001表示集合S-中拥塞节点的数目变化,2002表示集合S+中拥塞节点的数目变化,2003表示集合S<sub>0</sub>中拥塞节点的变化。图11展示了信号线集合S-和S+的拥塞节点数目、以及它们共同的拥塞节点数目的下降趋势。这些数据表明,共同的拥塞节点数目的比例低,并且其数目比其他的拥塞点数目更快地降到零,在实践中证实了本技术方案并行算法的收敛性。

[0085] 综上,本技术方案对并行布线器ParRouter/ParRouter+与现有的串行布线器在具有代表性的FPGA电路进行了布线性能测试。在拥有Intel Xeon E5-2430 2.20GHz处理器和32GB内存的4节点服务器上,布线器ParPouter在使用2/4/8/16/32的进程数目时依次获得的平均加速比是1.49倍/2.94倍/3.95倍/4.43倍/5.88倍;布线器ParPouter+在使用2/4/8/16/32的进程数目时依次获得的平均加速比是1.56倍/3.15倍/4.62倍/6.02倍/7.06倍。

[0086] 需要注意的是,公布实施例的目的在于帮助进一步理解本发明,但是本领域的技术人员可以理解:在不脱离本发明及所附权利要求的精神和范围内,各种替换和修改都是可能的。因此,本发明不应局限于实施例所公开的内容,本发明要求保护的范围以权利要求书界定的范围为准。

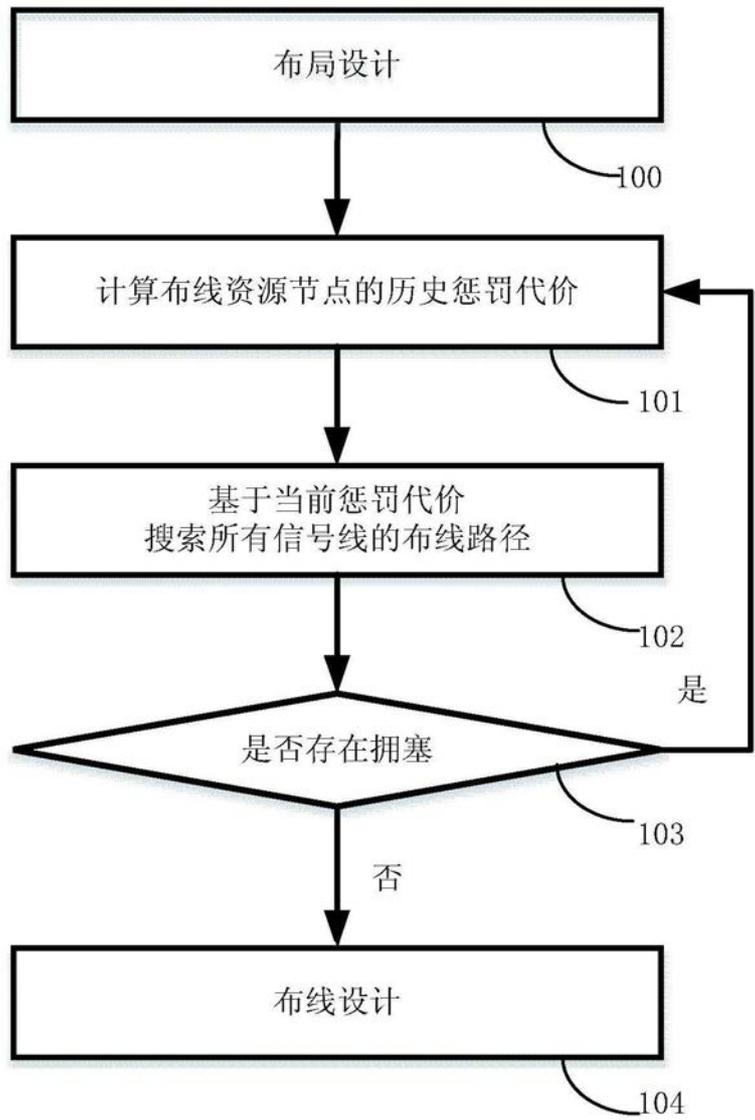
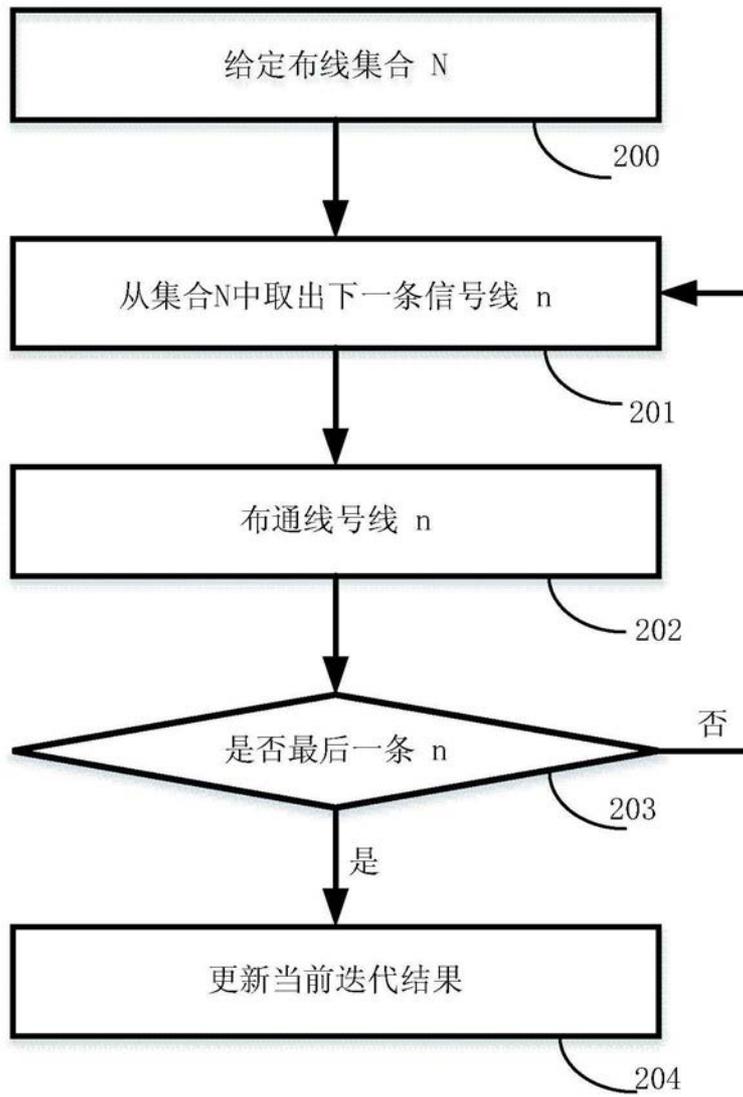
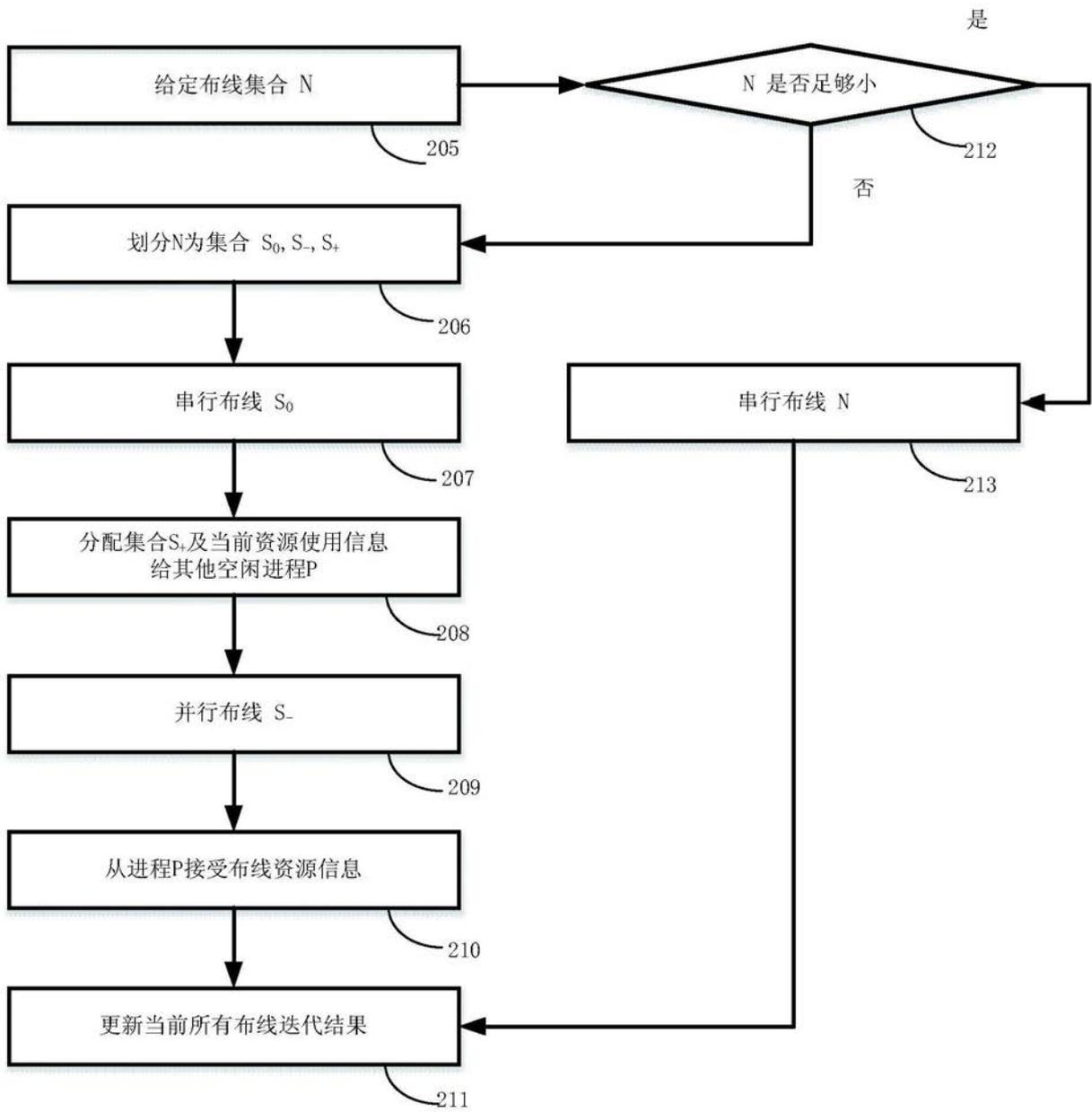


图1



2.1



2.2

图2

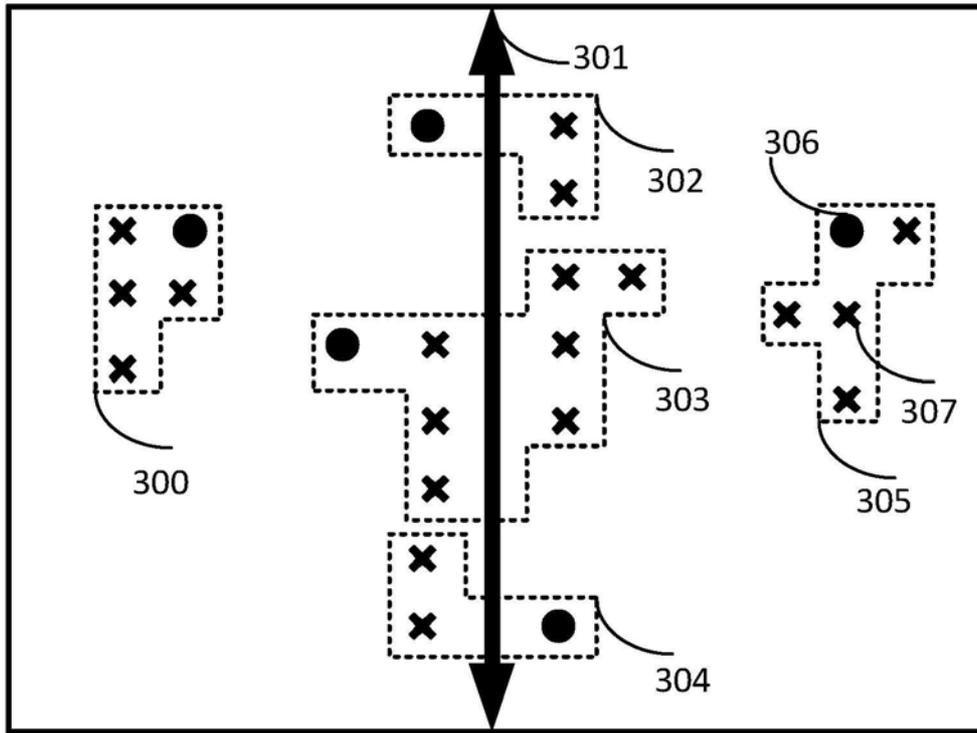


图3

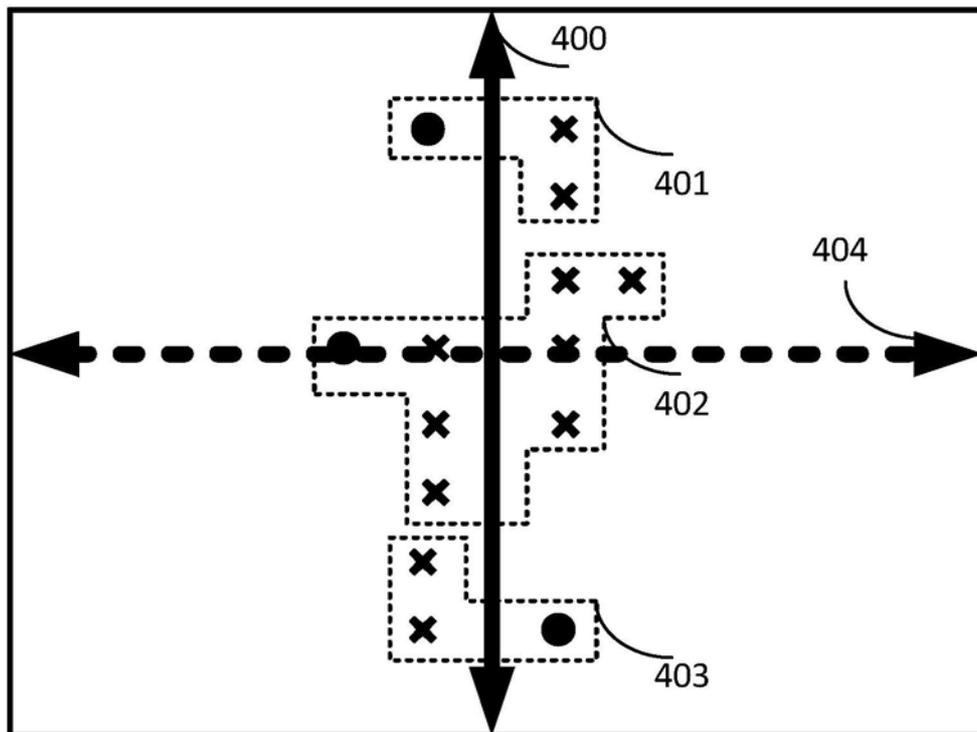


图4

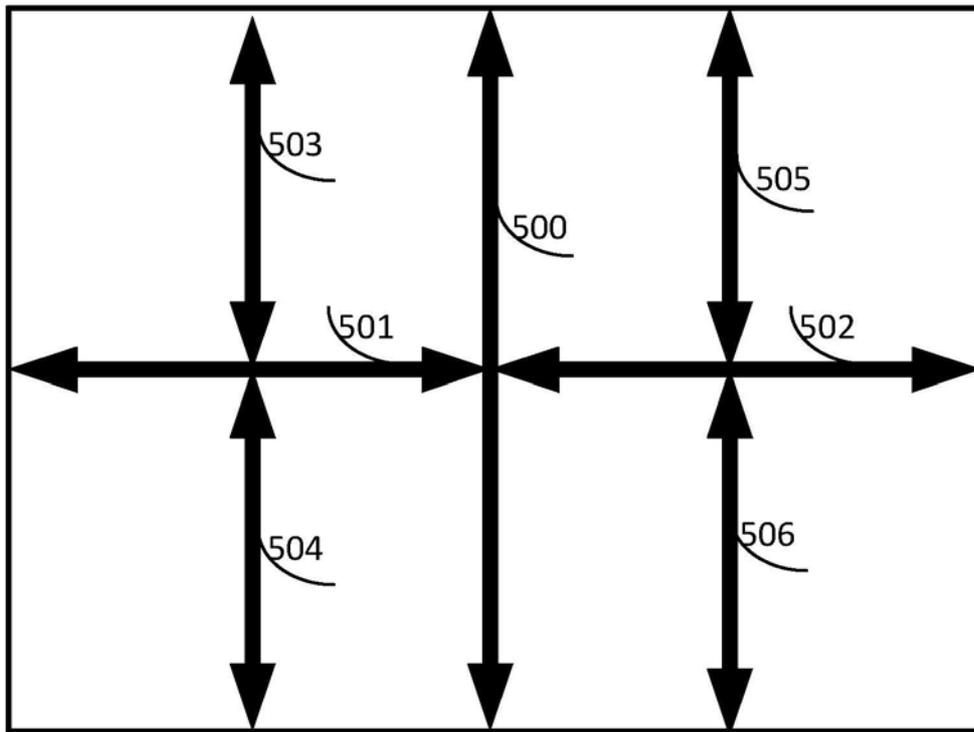


图5

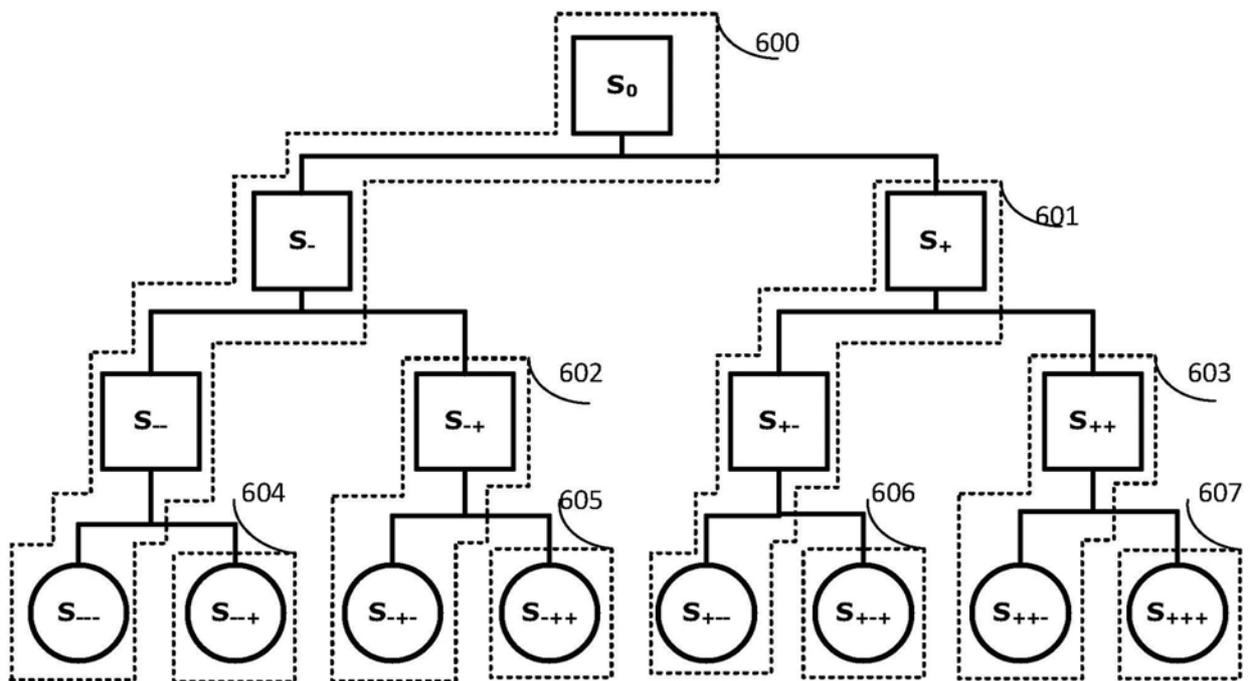


图6

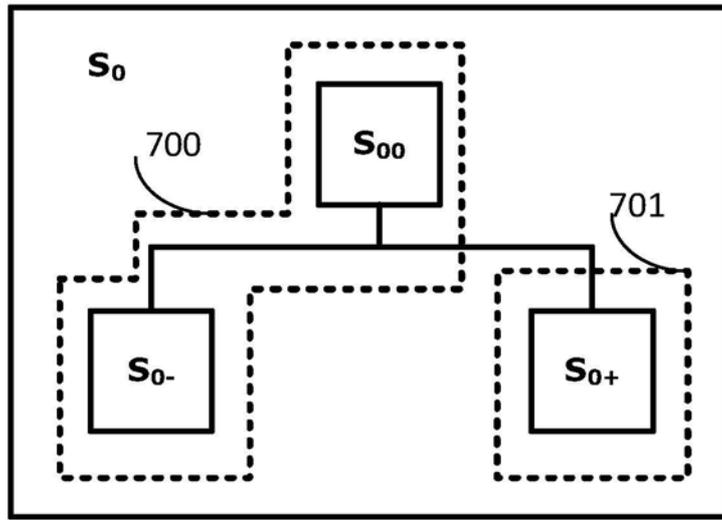


图7

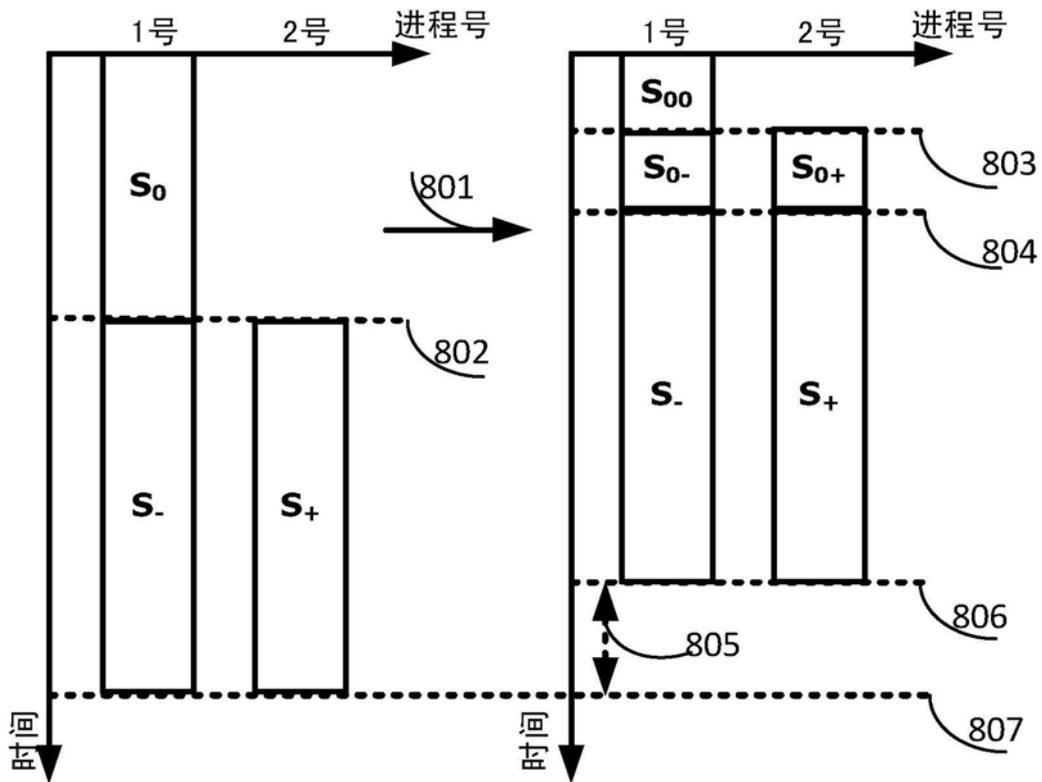


图8

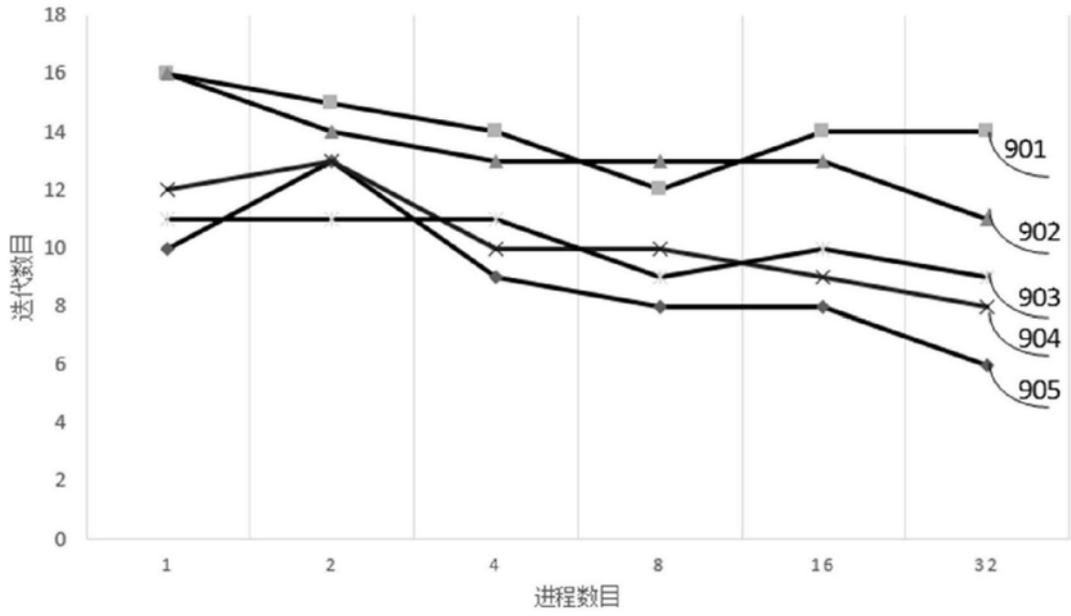


图9

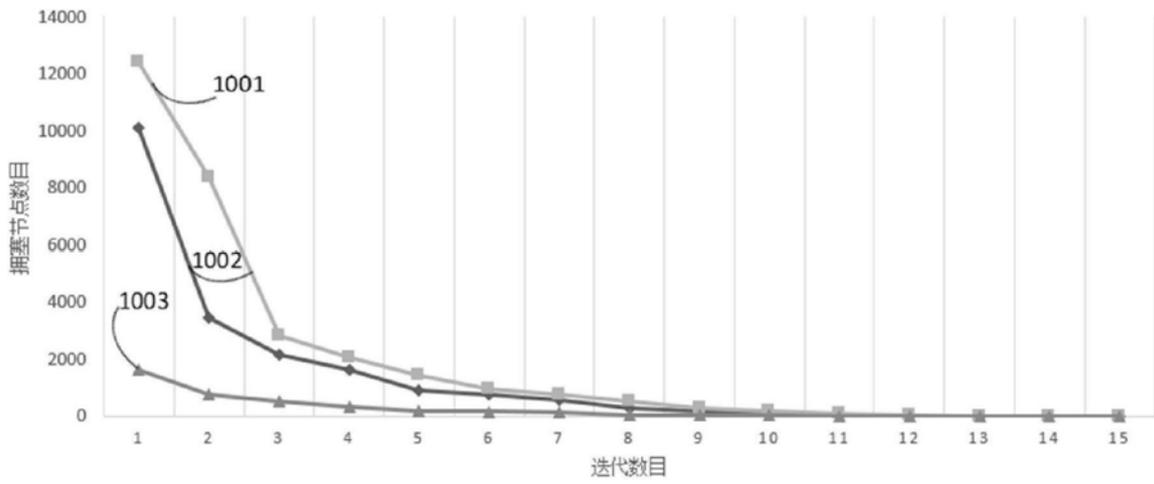


图10

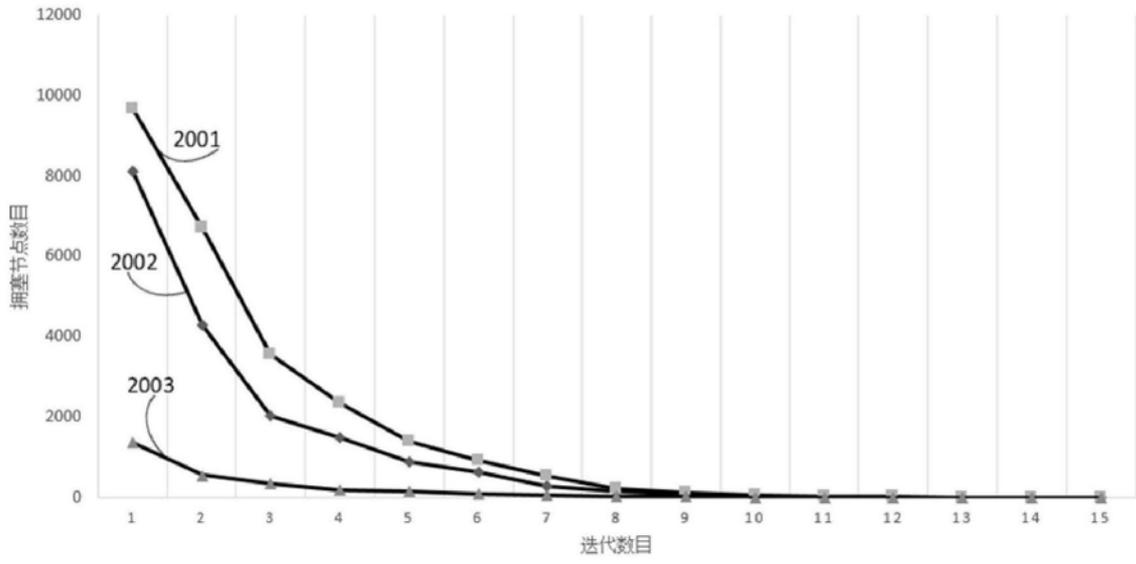


图11