第三章 基于斜线划分的平面化零时钟偏斜划分算法

在第二章中，已经对时钟树的拓扑结构生成算法作了简要的介绍，但是前面所提到的算法，有的基于垂直水平线划分例如 3M[3]方法等，可能无法满足一些比较随机的时钟接收端点分布，有些基于线长延时但是不能保证平衡性，有些不能采用 Elmore 等时延模型[6-9]，但是不能满足平面化布线的要求。如何适应大部分随机分布的时钟端点的电路，如何在平面化的情况下，兼顾二叉时钟树的平衡性，如何将欧几里的平面化的划分转化到成为曼哈顿平面化的划分，这些问题都将在本章得到解决。

本章主要介绍的是基于 DME[13]思想，以及用 Planar-DME 算法[14,15]和在 Planar-DME 算法基础上进行平衡性处理的拓扑生成算法，采用这种算法生成的时钟树能够保证平面化的零时钟偏差的时钟二叉树自顶向下的一次生成，在生成的同时，建立在欧几里的平面上的连线，并通过第二步进行自底向上的从欧几里的平面到曼哈顿平面的转换，产生一棵满足曼哈顿平面化的零时钟偏斜的时钟拓扑书及其连线。

§ 3.1 Linear-Planar-DME 算法介绍

Planar-DME 算法是 DME（Deferred-Merge Embedding）算法的一个扩展算法。DME 算法是一种线性复杂度的算法，采用线长模型，对于任何给定的拓扑关系都能够得到一个最优的嵌入，即零时滞和 $cost(GT(S))$ 最小。接下去，首先介绍 DME 算法。

§ 3.1.1 DME 算法

给定 sink 集合 S 和拓扑关系 G，通过以下步骤把每个内结点 $v \in G$ 嵌入到曼哈顿平面上：i）自底向上建立 *merging segment* 树，它代表了零时滞时钟树 T 中内结点可能的位置；ii)自顶向下决定 G 的内结点的准确位置。

在自底向上底过程中，每个结点 $v \in G$ 都有一条 *merging segment*，它代表了结点在最小 *cost* 零时滞树中可能的位置的集合。结点 v 的 *merging segment* 是由

它的两个子结点a和b的*merging segment*决定的。定义TS_a和TS_b是以a和b为根的*merging segment*子树。当TS_a和TS_b合并成T时，取保持时钟树零时滞的同时线长增加最小的点为结点v可能的位置。也就是说，从$plm(v)$到以v为根的子树的所有叶结点延迟（线长）相等的前提下，使T中的$|e_a|+|e_b|$最小。满足这个要求的$|e_a|$和$|e_b|$是唯一的，它们将被记录下来，并在自顶向下的过程中用来寻找嵌入点。

术语：

1) *Manhattan arc*：斜率为+1或-1的线段。同时我们把点看成是线长为0的特殊的*Manhattan arc*。如图3.1中ms(v),ms(u)为斜率为1，而ms(w)为斜率为-1的Manhattan arc。

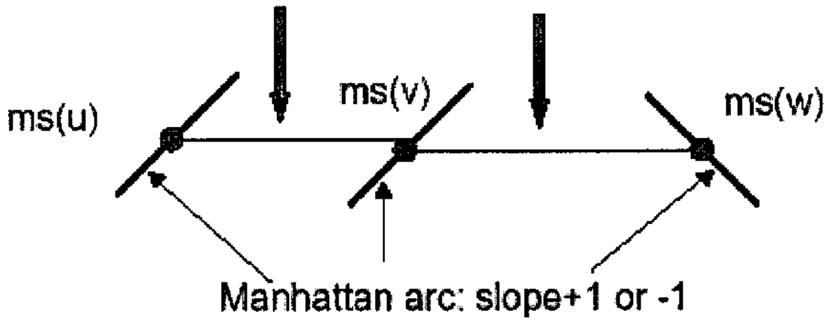


图3.1 Manhattan arc

2) *TRR* or *tilted rectangular region*：到某条*Manhattan arc*的距离小于等于固定值的点的集合，构成的一个边界分别由四条*Manhattan arc*组成的斜矩形。在*TRR*中心的*Manhattan arc*就称为*TRR*的*core*，而*TRR*的半径就是它的*core*到边界之间的距离。一条*Manhattan arc*本身是半径为0的特殊*TRR*。如图3.2所示

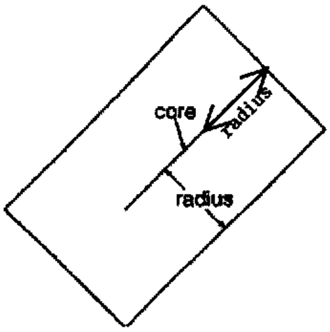


图 3.2TRR 示意图以及它的 core 和 radius

3) $ms(v)$是结点$v \in G$的*merging segment*，同样是一条 Manhattan arc。递归定义如下：如果v是已知时钟接收端点 s_i，则定义 ms(v)就是时钟接收端点 s_i；如果v是一个内结点，则$ms(v)$就是合并当前 v 结点的两个子结点 a 和 b 的子树TS_a和TS_b时增加线长最小的并且增加后总线长保持相等的位置的点的集合，也就是说，$ms(v)$是到$ms(a)$的距离不超过$|e_a|$，到$ms(b)$的距离不超过$|e_b|$的点的集合。（$|e_a|$和$|e_b|$分别为计算出来 TSa 和 TSb 相连后使得从当前父节点到底层叶结点也就是时钟接收端点相等的左右连线的最大长度）那么，$ms(v) = trr_a \cap trr_b$，它是通过相交trr_a（以$ms(a)$为*core*，半径为$|e_a|$的*TRR*）和trr_b（以$ms(a)$为*core*，半径为$|e_a|$的*TRR*）得到的。如图 3.3 所示

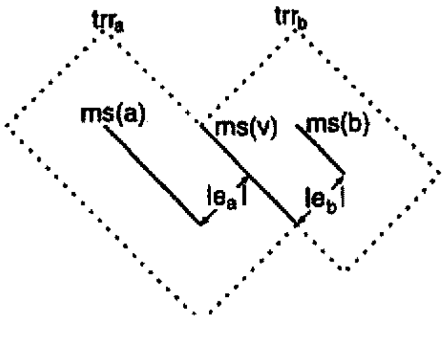
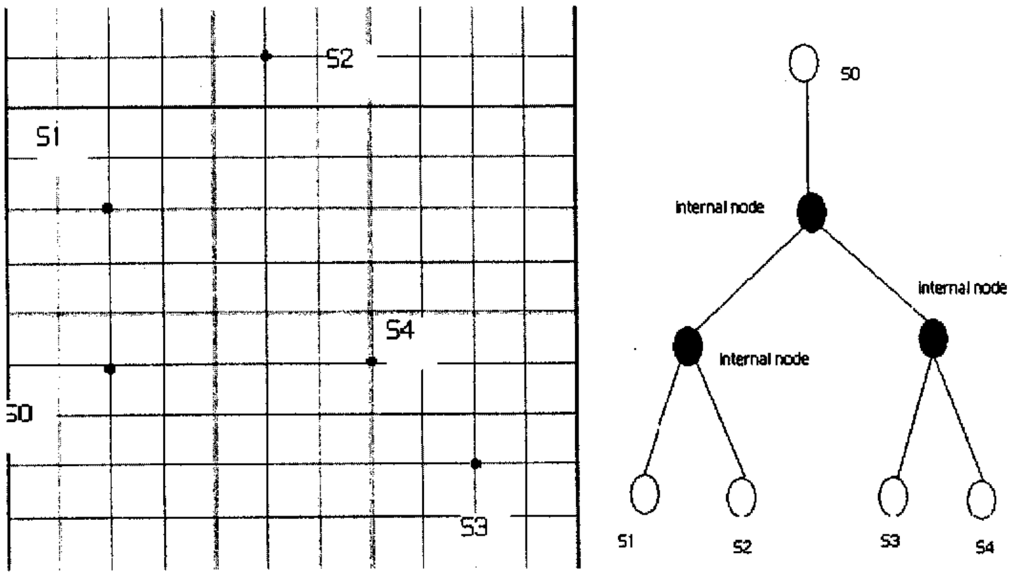


图 3.3 ms(v)的生成

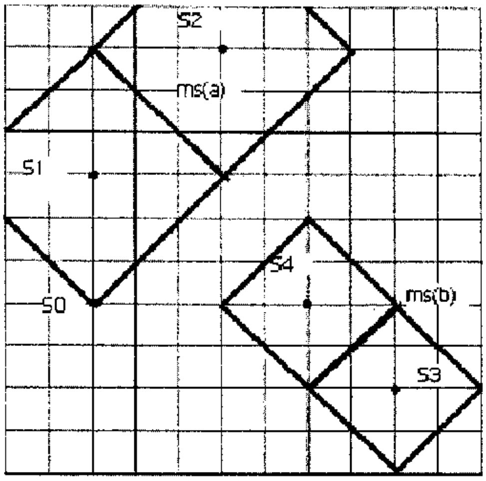
可以证明，如果$ms(a)$和$ms(b)$都是*Manhattan arc*，那么$ms(v)$也是*Manhattan arc*。因为$ms(s_i)$是一个点，也是*Manhattan arc*，所以所有的*merging segment*都是*Manhattan arc*。

得到与拓扑关系 G 对应的 *merging segment* 树后，就可以自顶向下确定 ZST 中各内结点的嵌入位置。算法如下：(i) 如果 v 是根结点，则取 $ms(v)$ 上的任何一点为 $plm(v)$，如果给定 *source*，那么取 $ms(v)$ 上离 *source* 最近的 *cost* 点。(ii) 如果 v 是除根结点外其他内结点中的一个，那么我们选择这样的一个点为 $plm(v)$，它满足：离 $plm(p)$ 的距离小于等于 $|e_a|$。也就是说，我们可以取 $ms(v)$ 和 trr_p（以 $plm(p)$ 为 *core*，半径为 $|e_v|$ 的 *TRR*）交集中的任何一点。

图 3.4 给出了一个 DME 算法步骤的实例。



(a) 一个给定的时钟接收端点拓扑结构和物理分布



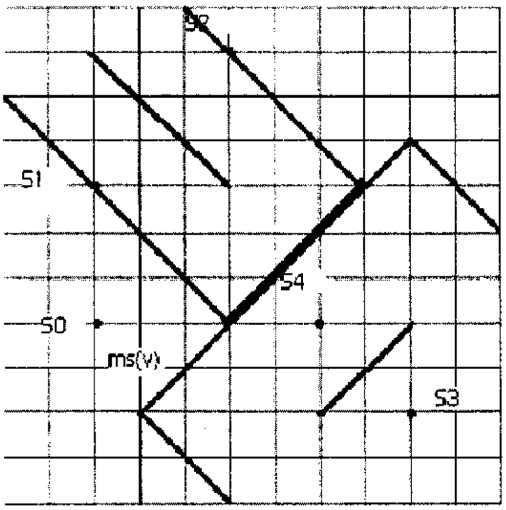
(b) 底层时钟接收端点的ms(v)生成

1） 首先完成s1和s2的连接：将s1和s2当做长度为0的Manhattan arc求出他们的ms(v)，分别以s1和s2为core，作出两个菱形的特殊TRR，不断同时增大TRR的radius，直到两个TRR相交。则两个TRR相交得到的线段就是所求ms(a)。同理得到s3和s4的ms(b)。

2） 我们得到从ms(a)到s1和s2的最大距离以及从ms(b)到s3,s4的最大距离

e(s1)=e(s2)=3

e(s3)=e(s4)=2



(c) 内结点的ms(v)生成

3) 得到ms(a)到ms(b)的最小距离d(a,b)= 7，因此连接两者的结点v到两者的分别的距离总和

$$e(a) + e(b) >= 7 \text{ ----------1}$$

而前面得到子树Ta和Tb的连线长度分别是

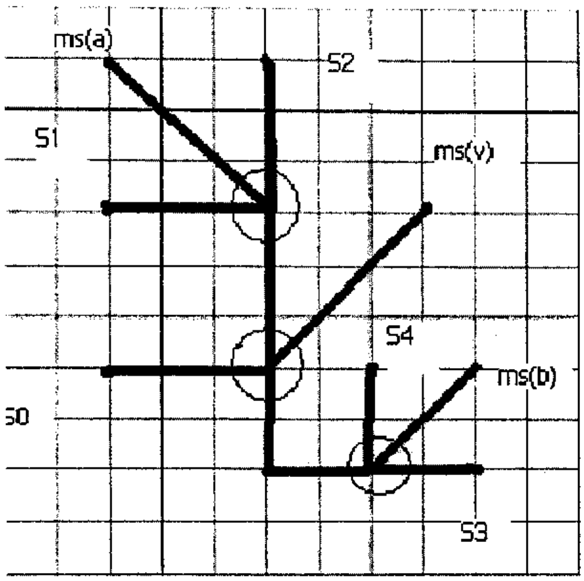
$$t(a) =3 \text{ and } t(b) =2$$

要使得Skew = 0 则

$$e(a) + t(a) = e(b) + t(b) \text{ ----2}$$

4) 从 1 和 2 我们得到 e(a)=3 and e(b)=4

画出相应的ms(a)和ms(b)的TRR并且找到它们的相交线段则得到ms(v)。



(d) 自上而下的连线生成

图 3.4 DME 算法实例

需要注意的是，DME 算法需要一个给定的拓扑关系作为输入。对于不同的拓扑关系，用 DME 算法建立的时钟树的 *cost* 是不同的，因而有许多产生低 *cost* 的时钟树拓扑关系的算法。DME 算法建立的时钟树不满足欧几里德平面性。

§ 3.1.2 Linear-Planar-DME 算法

定义：

1）$diam(S')$、$radius(S')$：对任意的 sink 子集 S'，定义 $diam(S') = \max\{d(s_i,s_j)\,|\,s_i,s_j \in S'\}$ 为 S' 的直径，

2）$radius(S') = diam(S')/2$。

3）$Manhattan\ disk$：以点为 $core$ 的 TRR。

4）$MD(s_i,r)$：以 $\{s_i\}$ 为 $core$，以 $r \geq 0$ 为半径的 $Manhattan\ disk$。

5）$center(S')$：假设 sink 子集 S' 的半径为 $radius(S') = r'$，则中心 $center(S') = \cap_{s_i \in S'} MD(s_i, r')$，该点到 S' 中任何一个 sink 的距离不超过 r'。

6）$c(S')$：$center(S')$ 的中点。

7）$P_{S'}$：包含 S' 的欧几里德凸多边形。