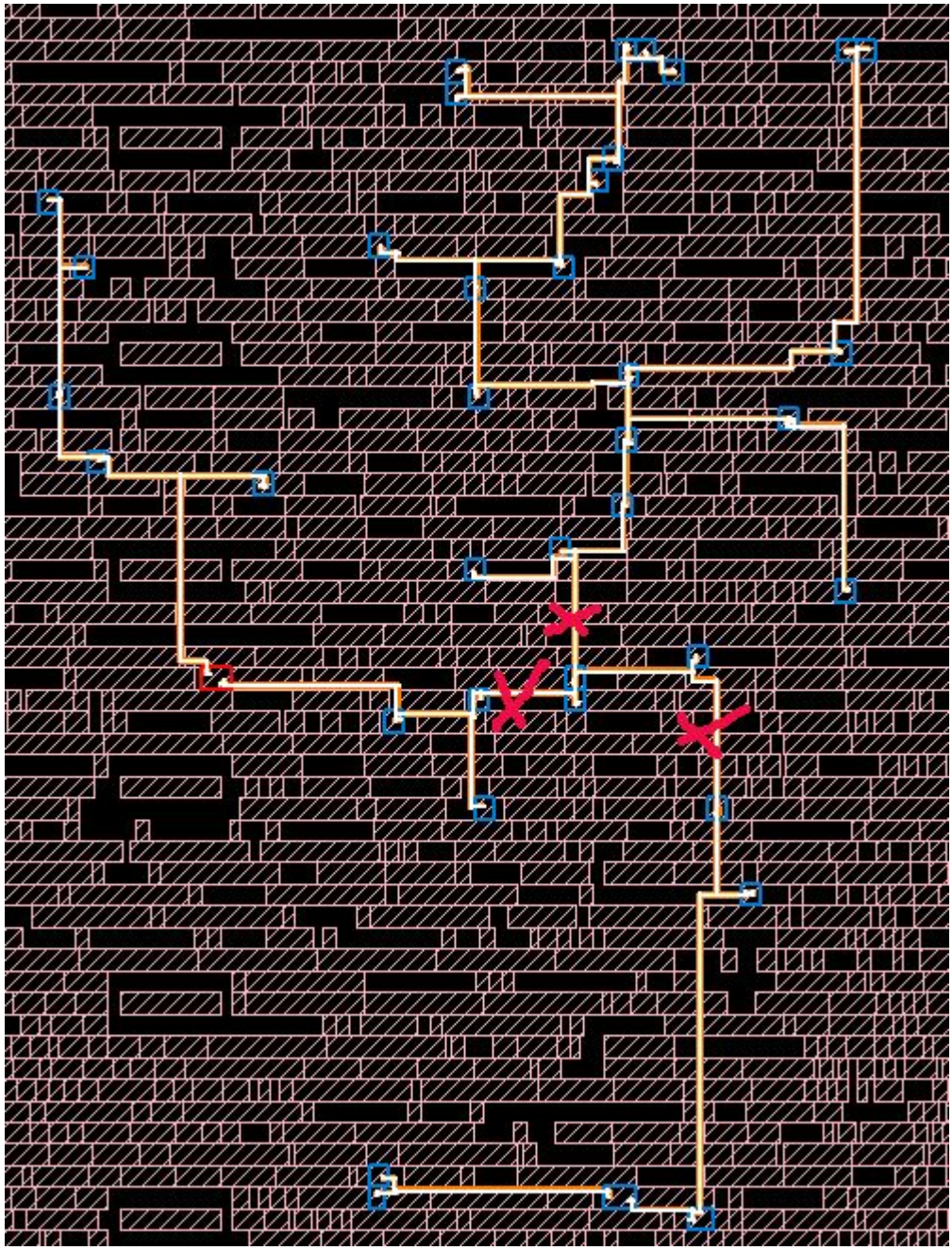
Manual Add Buffer on Route AN

一、这次先说大家关心的----怎么使用。Too simple and user friendly。（注意我们是在讨论插 buffer， 不是 size_cell, size_cell 实在太简单了）

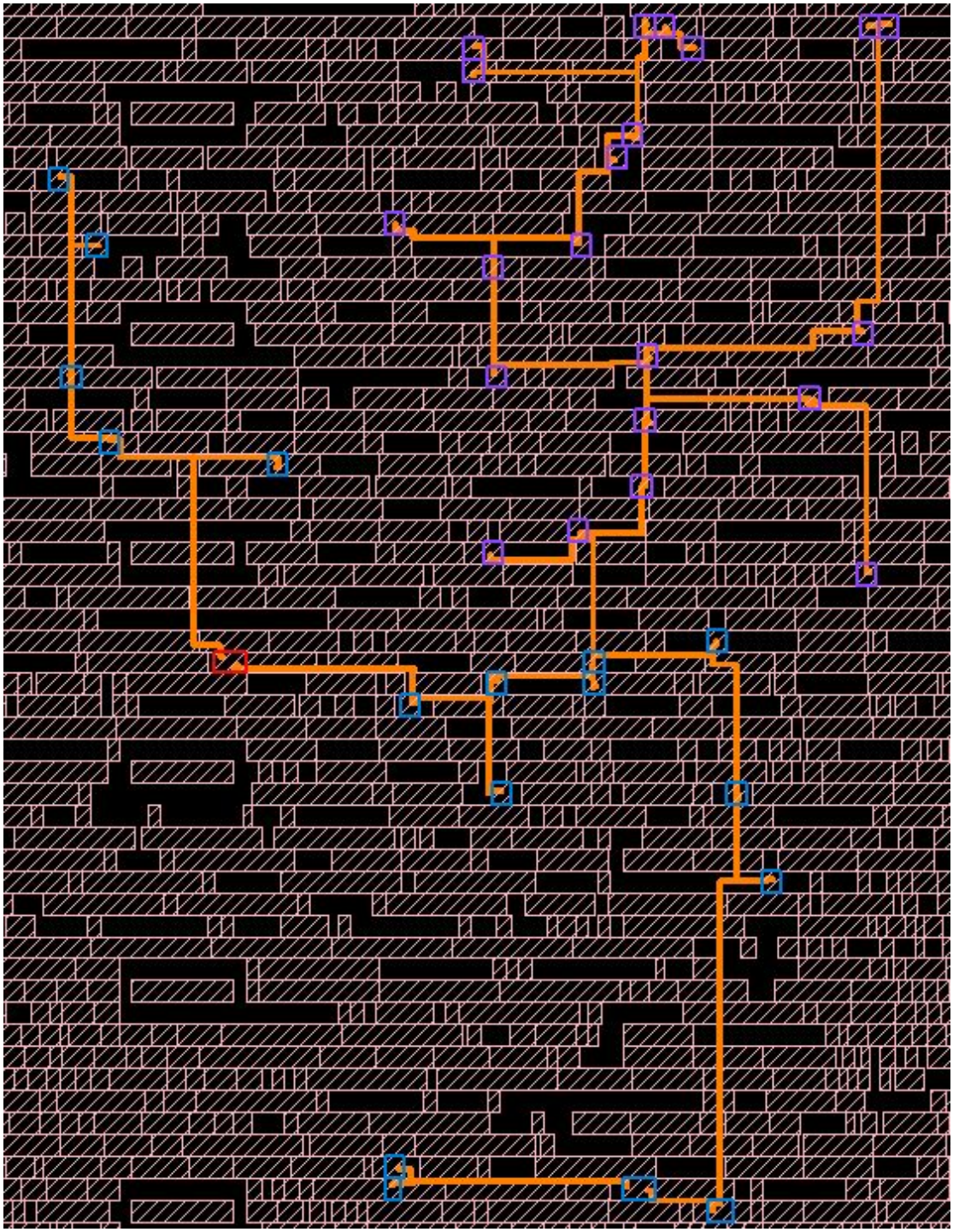
3 部曲---- 一看二插三 next (你只需点鼠标就可以了). A picture is worth a thousand words， 看图



假设你现在坐在 ICC 面前，如上图 红色的框框是 drive cell 蓝色的框框是 load cell， 黄颜色的线条是 net，这些都是自动的帮你高亮出来了。然后现在这条 net 有 transition violation，你要手动修，我想你一眼就能看出来在哪插 buffer 吧。 我用红色叉叉标记的地方都是不错的选择-----有放 buffer 的

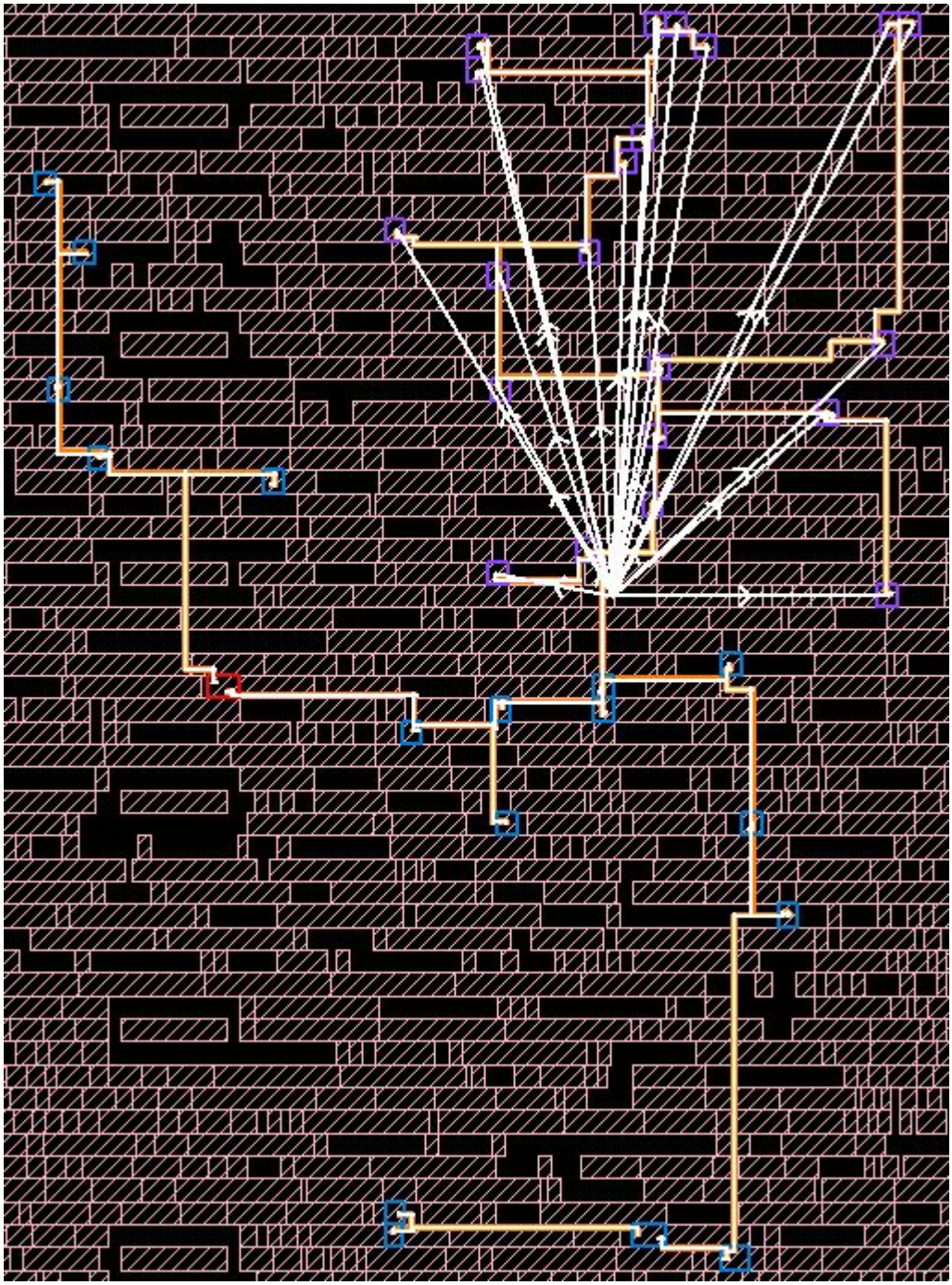
坑而且可以将原来的线长减小很多。

一看： 先假设我们选择在上面那个红色叉叉那里插 buffer，在靠近红色叉叉那里按住鼠标中键，由左上方往右下方画一条 45 度短线段穿过叉叉附近的那条 net shape 告诉 ICC 你要在那里插 buffer 啦，ICC 理解你之后，作出反应如下图



紫色的框框将是你插 buffer 之后， 由该 buffer 来驱动这些 cell，蓝色的框框不变还是有原 drive cell 来驱动， 真正意义上的 add buffer on route 吧。

二插：（这一步本可以和上一步合并，鉴于脚本还未进行大量的测试，稳定性起见增加这一步）如果 ICC 没有理解错你的意思的话， 任意位置单击鼠标中键叫 ICC 在刚才那红色叉叉那里插个 buffer 吧。 如下图



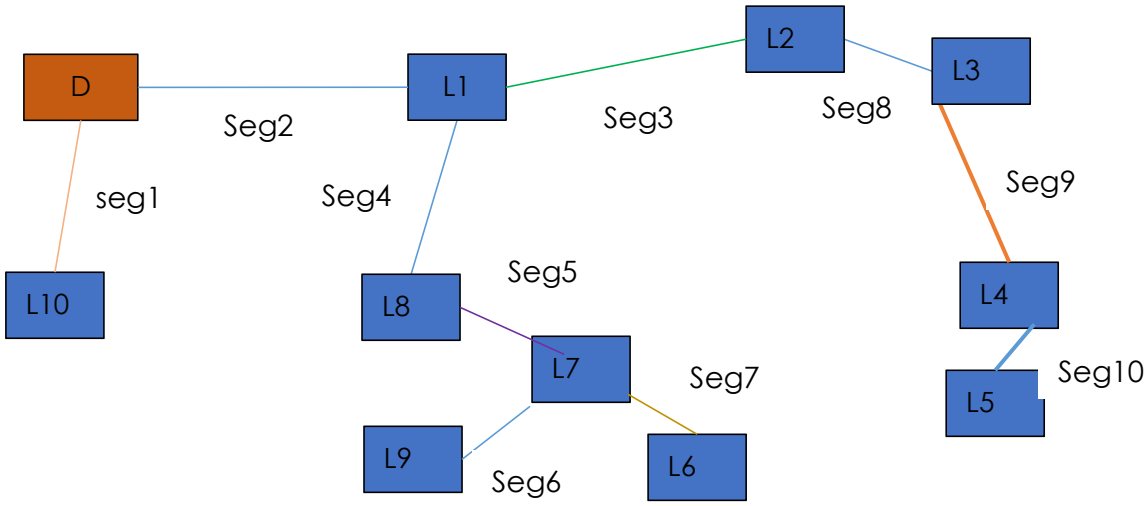
可见紫色的框框都由新加入的 buffer 来驱动了。

三 next：(这一步是依据编程语言（特别是面向对象语言）里的 iterator 概念而设计的，只不过我在这里用 TCL 来简单的实现 iterator) 上一条 net 修掉了是不是？好吧，任意位置按住 shift 键的同时单击鼠标中键高亮到下一组 net drive load 回到一看开始修下一个 transition violation。

就这么些， easy 吧。

二、来看一下怎么实现

先来做一道小学题， 看图说规律



这是一幅 ICC 绕线后的抽象图，D 为 drive cell 驱动 10 个 load L1, L2, ..., L10，现对每一条线段 segment，跟随电流的方向（取由 D 流向各 load 的情况），列出每条线段后面挂了多少个 load。

线段名	Load 名
Seg1	L1
Seg2	L1,L2,L3,L4,L5,L6,L7,L8,L9
Seg3	L2,L3,L4,L5
Seg4	L6,L7,L8,L9
Seg5	L6,L7,L9
Seg6	L9
Seg7	L6
Seg8	L3,L4,L5
Seg9	L4,L5
Seg10	L5

假如我在线段 seg3 上插 buffer， 从图中可以看出 L2,L3,L4,L5 将由新插的 buffer 驱动，其余 load 不变。在其他 segment 上插 buffer 类推。

再回到上面那张表我们发现，如果我们能够自动的（而不是人工）得到这张表（线段与 load 的对应关系）， 那么 add buffer on route 来修 transition 将会易如反掌，这是整个设计的核心。

产生这张表有高逼格的算法，鉴于其复杂度和难度，我这先用土鳖的方法，实在不行再考虑用高逼格的算法。 当然，二者思想是一样的。

回到我们一开始说的具体操作上，有那么一个操作是用鼠标画一条线段将 ICC 绕线后的 net shape 分成两部分，很显然其中一部分是包含那个 drive cell 的，另外一部分是只包含部分 load cell 的（这些 load cell 正是我们要插 buffer 的目标），仔细想想是不是？（比如现在将 seg3 打断）

实现到此结束。

QA

Q: 我可以选择不同大小的 buffer 吗？
A: 抱歉，我不实现这样的功能。我个人的理由是：工具残留的 transition violation 就那么几十条到撑死百来条，直接拿大的 buffer 开搞，何必纠结于这种小细节，而且最佳 buffer 是由工具仿真计算出来的，不是人 YY 出来的。
如果你一定要这个功能，我建议你弄个 ICC 悬浮菜单（据我所知，公司有人写好了）来动态更改脚本里的 buf_ref_name 这个变量，我这里就不重复造轮子了。

Q:某些长线插一个 buffer 可能不够，需要连续插多个，支持吗？
A: 回答是肯定的，找到你要插的位置继续点鼠标就可以了。如果你理解了上面那张映射表，你就会知道插没插 buffer 对你后续插 buffer 是一码事。比如我们在 seg3 上插了 buffer 的话，那就动态更新这张表，即现在 seg3 后面只挂了你刚插的那个 buffer， 接着你可以比如在 seg2 上继续插 buffer，是吧。

Q: 有些 transition violation 我想通过 size_cell 修，咋整？
A: 参考第一问的答案。另外写好脚本在命令行里搞也是很快的

Q: 万一 ICC 显示的要插 buffer 的 cell 和实际的不符呢（即有 bug），怎么办？
A: 是的，肯定会有这种情况发生。毕竟很难做到完全通用。这时就需要你显示的指明那些 cell 了。这里我建议你去看一下 set_gui_stroke_binding 的 manpage，上面说了它支持你画一个框以及各种复杂的路径。那么你就可以画一个框框住你要插 buffer 的 cell 告诉 ICC， 这比某插件菜单来的更快。这个我这边不实现，有兴趣的同学可以当作是个小练习，很简单的。

Q: 虽然你这个方法很快，但我还是想脚本自动的修，手修累，有自动的么？
A: 矫情！$%#@%^#$!。需要牛逼算法以及高级数据结构，一个人的话工作量太大了。

<完>
2016/01/23