第三届（2021）集成电路 EDA 设计精英挑战赛
赛题指南

一、**赛题三**：海量图形数据的高速区域化查询

二、**命题企业**：南京集成电路设计服务产业创新中心有限公司

三、**赛题 Chair**：杨　帆　　复旦大学微电子学院 教授

四、**赛题描述**：

1. 总体描述

(1) 定义核心区域(core area) 为一个坐标区域，如{{0, 0}, {4000, 2000}}

(2) 核心区域内存在大量的矩形图形（正多边形已转换为最大化矩形），每个矩形图形为一个小的坐标区域，如{{10, 10}, {50, 30}}, {{511, 320}, {517, 360}}...

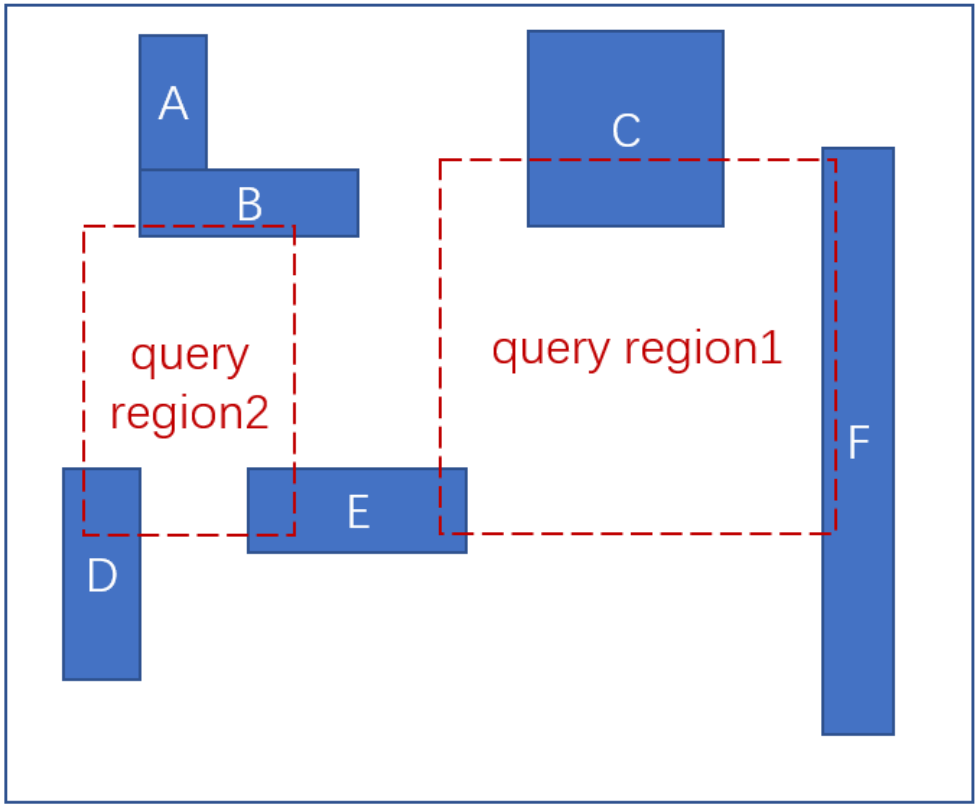
(3) 定义查询区域(query region)为坐标区域，如{{50, 50}, {100, 120}}，快速获取该区域内的矩形图形

例1， 如下示例图，核心区域有物体 A、B、C、D、E、F 六个矩形。

如果在”query region1”进行区域查询，则得到 C、E、F 三个矩形（查询结果不分先后）。

如果在”query region2”进行区域查询，则得到 B、D、E 三个矩形（查询结果不分先后）。

如果查询区域为整个核心区域，则六个矩形 A、B、C、D、E、F 全部被得到。

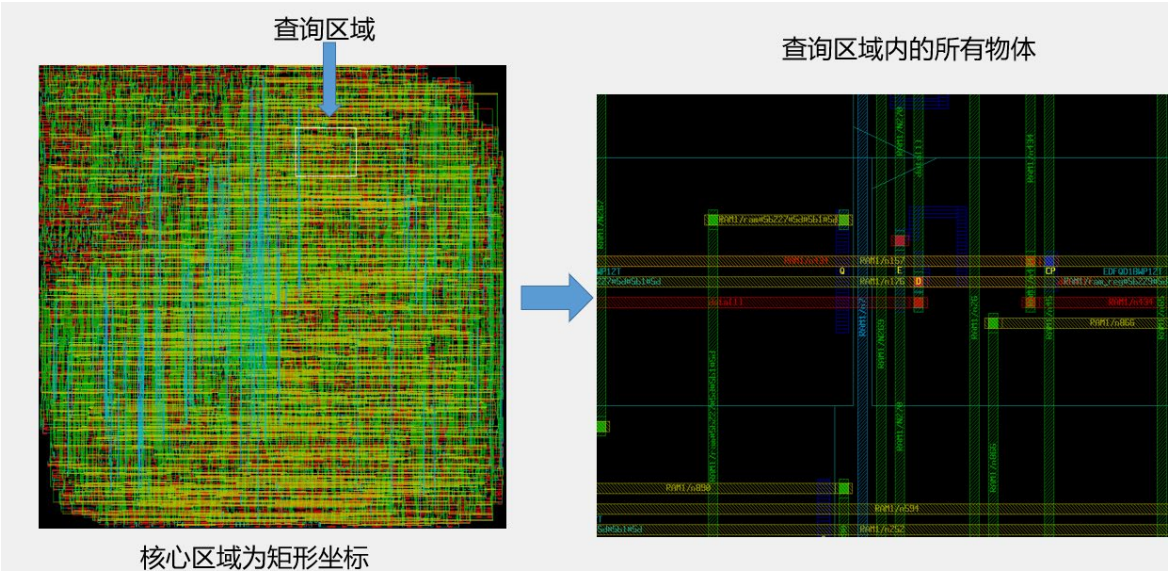


core area

例2， 如下版图，核心区域是一个布线后的设计，其中包括 IO pin 和 routing blockage, instances(pin/obs), power stripes(wire/via)以及 signal routing(wire/via)。

指定下图（左）的查询区域，将得到该区域的所有物体(pin/obs/wire/via)，即下图（右）中显示物体。

注：版图中的物体皆可看作一个矩形。



五、评分细则

1. 公开测试用例评分（60 分）

分数（30 分）	正确性	运行时间(CPU time)	使用内存(peak mem)
项目构建	/	5	4
查询区域1	3	2	2
查询区域2	3	2	2
查询区域3	3	2	2

用例 1 和用例 2 各 30 分，评分标准如上表格，分数为每项可得到的标准分。

1.1 正确性：查询结果与 golden 结果一致得满分，不一致得 0 分且运行时间和使用内存得分为 0

1.2 运行时间：得分=(golden 运行时间/参赛程序运行时间) * 标准分

1.3 使用内存：得分=(golden 使用内存/参赛程序使用内存) * 标准分

注：

注 1：运行时间和使用内存的得分精确到小数点后两位。总分评比也精确到小数点后两位。

注 2：得分可能高于标准分。

注 3：EDA 创新中心会在统一的环境下运行多次取平均值（以减小误差）。Golden（及其运行环境）仅供参赛选手参考。

2. EDA 创新中心测试评分（30 分）

2.1 测试用例由 EDA 创新中心提供（case 较大，内存和运行时间较大）

2.2 EDA 创新中心运行选手程序得到运行结果并根据结果评分

2.3 评分标准与公开测试的评分标准一致

3. 源码及文档评分（10 分）

3.1 设计文档（算法设计，数据结构设计）评分（4）

3.2 源代码质量（编码风格，SEGV，Memory Leak）评分(4)

3.3 测试报告(单元测试，系统测试）评分(2)

注：因此项评分主观性较强，所以设置分数值较低。

六、运行环境

1. 开发环境配置

资源路径：https://gitee.com/niiceda/open-edi/tree/open-edi-contest/

推荐开发环境：ubuntu 20.04（或 centos 7.9.2009）

- 已有 Linux 环境手动配置：参考源码库中 config_guide_ubuntu20.04.md（或 config_guide_centos7.9.md）
- docker 方式：使用资源库中 fast_deploy_ubuntu_20.04.dockerfile 构建开发环境镜像

以上环境准备妥当后，启动相应环境

2. openeda 下载、编译运行

mkdir $yourSandBox

cd $yourSandBox

source /opt/devtoolset/devtoolset.bash

git clone **-b** **open-edi-contest** --recursive http://gitee.com/niiceda/open-edi.git

cd open-edi

mkdir build

cd build

cmake .. -DCMAKE_PLACE=0 (for opt) or cmake ..-DCMAKE_PLACE=0 -DCMAKE_BUILD_TYPE=Debug（for dbg， apt install gdb for debug command）

make -j 4

export LD_LIBRARY_PATH=/usr/local/lib

$yourSandBox/open-edi/build/src/app/openeda

OpenEDA%

注 1：www.openeda.com，开源网站访问地址。

注 2：openeda 及其运行环境提供了统一的软件平台，为评分提供了统一的标准。如果参赛选手使用其它软件平台(如 Windows 操作系统、python 语言等)，EDA 创新中心只能尽量提供相近的评分标准和运行环境，以保证公平性。

注 3：EDA 竞赛组委会提供统一的服务器环境，EDA 创新中心会在该系统配置 openeda 的运行环境，供参赛选手测试。

七、测试用例

1. 用例 1

参看 open-edi-contest 下”**demo/contest/test_case1.zip**”，文件包括：rtk-tech.lef, stdcells.lef, route.def , query.tcl, region1_query_result.txt, region2_query_result.txt, region3_query_result.txt。运行如下：

$yourSandBox/open-edi/build/src/app/openeda

OpenEDA% read_lef rtk-tech.lef

OpenEDA% read_lef stdcells.lef

OpenEDA% read_def route.def

OpenEDA% init_query

INFO(run time and memory)

OpenEDA% query $area1

Query result ：

INFO(run time and memory)

OpenEDA% query $area2

Query result ：

INFO(run time and memory)

OpenEDA% query $area3

Query result ：

INFO(run time and memory)

OpenEDA% cleanup_query

查询结果正确性可与 golden 结果对比（如果结果较多，请用 sort 排序后保存到文本文件，对文本文件进行 diff 比较）。

三个查询区域分别是：

area1: 77581 159239 79891 160882

area2: 182106 181587 276864 249013

area3: 69341 79441 279731 229145

2. 用例 2

参看 open-edi-contest 下" **demo/contest/test_case2.zip**"。

运行过程同用例 1。

三个查询区域分别是：

area1: 380015 434995 416981 461299

area2: 1435638 1759660 1543493 1836405

area3: 368731 591718 1565088 1442994

八、示例代码

1. src/db/rq 框架

data_model.cpp, DataModel 为导入 wire/via/pin/obs 到 LRect 中

data_model.h

rq.cpp，参赛选手程序入口

rq.h

rq_tcl_command.cpp，命令注册

rq_tcl_command.h

2. 接口

initQuery()

query(const Box &search_area)

cleanupQuery()

参看注释行“// add your code here...”

输入为 DataModel::geometries_(vector<LRect>)，输出为查询结果、运行时间和使用内存

3. 声明

3.1 参赛选手可根据自己需要增删改，包括文件和代码。数据结构和算法由选手自行设计和实现，但不可复制或使用 openeda 中的代码。

3.2 LRect 为示例，参赛选手可自行设计相关数据结构以及内存管理方式

3.3 DEBUG_QUERY 编译条件供参赛选手参考示例

3.4 选手可根据 layer_id 来存储数据，layer_id 最大值可设置为静态值 128

3.5 可多线程运行，多线程运行比较时间为 elapsed time（而不是 CPU time）

3.6 Monitor 用于监控内存和运行时间

3.7 Message 用于打印信息到控制台和 openeda log 文件

一、 参考文献

1. https://blog.mapbox.com/a-dive-into-spatial-search-algorithms-ebd0c5e39d2a
2. https://www.researchgate.net/profile/Wendy-Osborn/publication/306374828_Continuous_Region_Query_Processing_in_the_mqr-tree/links/5995fa69a6fdcc35c6bfeddb/Continuous-Region-Query-Processing-in-the-mqr-tree.pdf
3. http://www.cs.utah.edu/~lifeifei/cis5930/kdtree.pdf