Calibre 学习总结

第一章 Calibre 简述

1. 1 Calibre 简介

Calibre 作为Mentor Graphics 公司出品的后端物理验证（Physical Verification）工具，它提供了最为有效的DRC/LVS/ERC 解决方案，特别适合超大规模IC电路的物理验证。它支持平坦化（Flat mode ）和层次化（Hierarchical mode）的验证，大大缩短了验证的过程；它高效可靠的性能已经被各大Foundry 认证，作为Tape Out 之前的验证标准。它独有的RVE（Result ViewEnviroment）界面可以把验证错误反标到版图工具中去，而且良好的集成环境便于用户在版图和电路图之间轻松转换，大大提高了改错的效率。xCalibre 具有版图寄生参数抽取的功能。

1．2 手册

在工作站下输入 mgcdocs &命令，就可阅读 Calibre 的所有手册。

1．3 几个常用的缩写命令

1、 SVRF---Standard Verification Rule Format（标准的检查文件）

2、 RVE---Results Viewing Environment(显示结果用的环境窗口)

3、 SVDB---Standard Verification Database (LVS results)

4、 DRC---Design Rule Checking

5、 LVS---Layout Versus Schematic

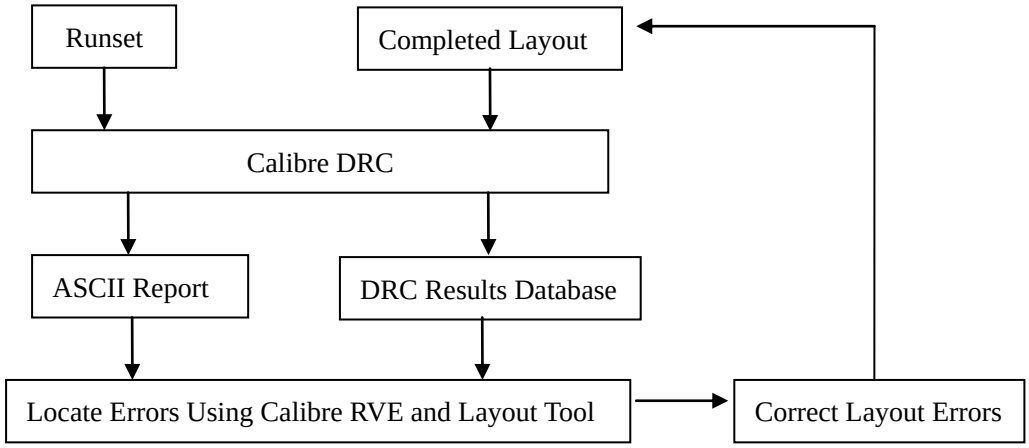
6、 ERC---Electrical Rule Checking

第二章 Calibre DRC

2． 1 数据准备

完成CalbireDRC需要的数据有版图数据和执行DRC检查的命令文件（Runset）。版图数据支持GDSII、CIF、BINARY、ASCII 格式。

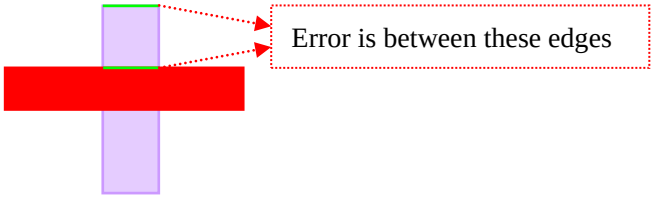
2．2 流程图



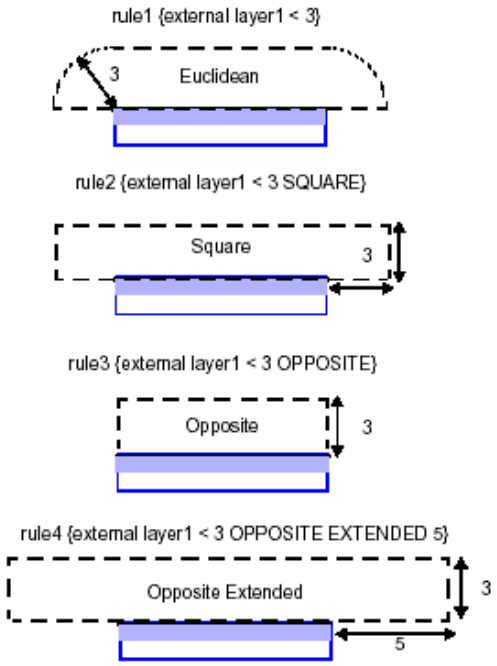
2．3 DRC Runset File

1 基本控制，原有 DRACULA 的 file 可以用 drac_cvt　sourcefile　targetfile 命令来转换。

（1）Calibre 是一个“Edge－Based”Tool，默认错误的显示是边



(2)DRC 检查的结果有三种控制 Euclidean（default）、Square、Opposite。



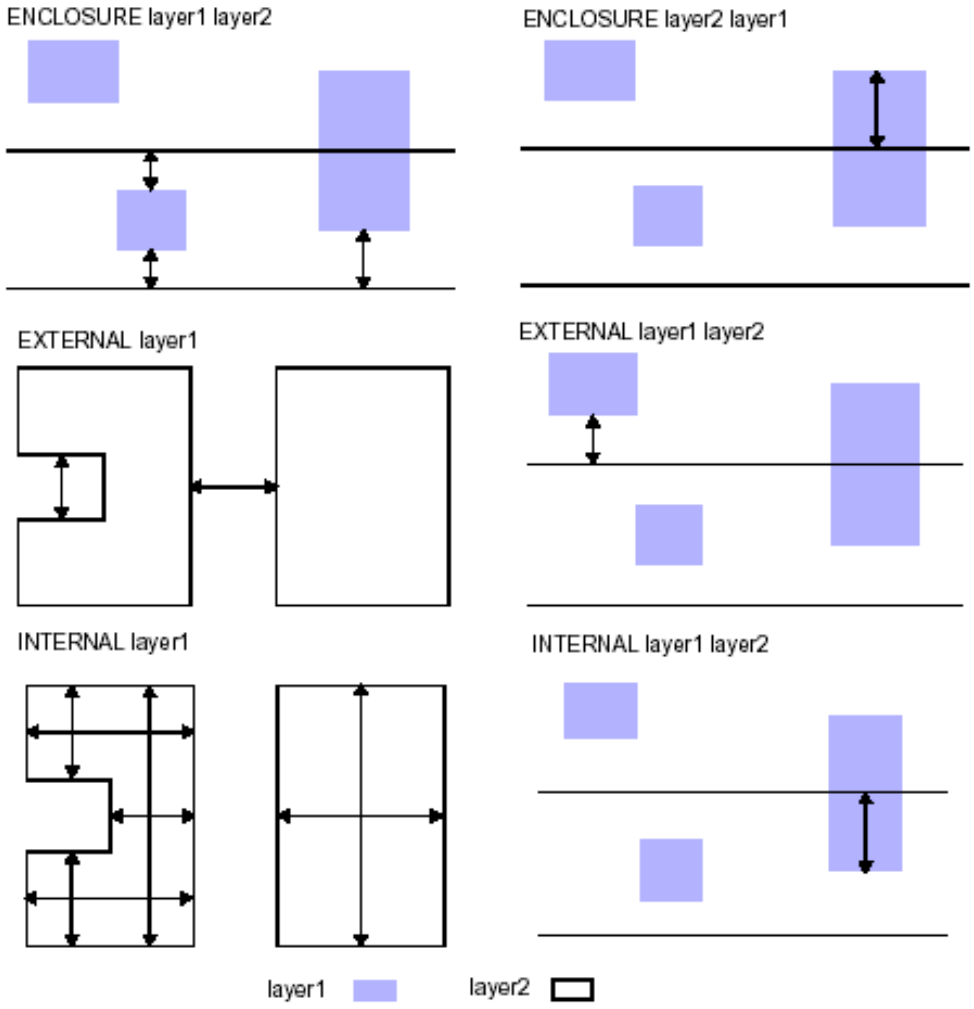
(3)常用的几条检查规则，具体可阅读 Calibre 的手册

(a) Internal（内边对内边）用来检查 Width、Overlap；

(b) External（外边对外边）用来检查 Space、Notch；

(c) Enclosure（内边对外边）

Figure 4-5. Measured Edges in the Dimensional Check Operations



2 一般的 DRC 检查文件包含以下几个部分：

(1) 运行设置，设置 GDS 的位置，结果文件放的位置等；

(2) 层次定义，定义输入的层次；

(3) 层次运算，产生运算需要的一些中间层次；

(4) 规则检查，具体对每条规则的检查；

(5) 选择控制，可以只检查某几条规则或者只检查某个单元。

3 一个简单的 Rule File，“//”后面为注释

```
//————————————————————————————————
     LAYOUT PATH   “/home/*…….”   //GDS 的路径
     LAYOUT PRIMARY   “*” //GDS 的顶单元名
     LAYOUT SYSTEM GDSII //版图数据的格式
//————————————————————————————————
    DRC MAXIMUM RESULTS 500 //最大错误输出数目
    DRC RESULTS DATABASE “/home/*……” //存放 DRC 错误数据的路径及名称
    DRC SUMMARY REPORT “/home/*……” HIER //存放 DRC 简述文件的路径及名称
//————————————————————————————————
    DRC CELL NAME YES CELL SPACE XFORM //表示底层的错误将直接在底层显示
    PRECISION 1000
    RESOLUTION 10 //两句合起来表示检查的数据格点为 10/1000＝0.01
//————————————————————————————————
    FLAG OFFGRID    YES //在 Summary Report 里显示不在格点上的版图的坐标
    FLAG SKEW       YES //在 Summary Report 里显示不是 45 度线的版图的坐标
    FLAG ACUTE      YES //在 Summary Report 里显示锐角的版图的坐标
//—————————————————————————————————
     //输入层次的定义
    LAYER nwelli    1
    LAYER ndiffi    2
    LAYER pdiffi    3
          .
          .
          .
   //TEXT 层及连接性的定义
   TEXT LAYER 60 ATTACH 60 met1
   TEXT LAYER 70 ATTACH 70 met2
   //定义连接
   CONNECT met1 poly1 BY cont
   CONNECT met1 ndiff BY cont
   CONNECT met1 pdiff BY cont
   CONNECT met2 met1 BY via1
   CONNECT met1 allnsub BY cont
   SCONNECT allnsub nwelli
   //顶层数据的定义，有利于提高检查速度
   LAYOUT TOP LAYER pad met2 via1

//———————————————————————————————
//层次的运算
 fpoly   =   poly1 INTERACT celiso
 tpoly   =   poly1 NOT INTERACT celiso
 pcode    =   SIZE pcodei BY 0.15 OVERUNDER
                    .
```

```
                .
                .
//一般用来层次运算用的命令大概有：INTERACT  表示有任何重合关系的，
//相当于 DRACULA 的 OVERLAP。NOT ,AND,OR,SIZE    同 DRACULA。SIZE  后面
//加上 UNDEROVER  表示先缩小再放大，OVERUNDER 表示先放大再缩小。
//wmet1 = WITH WIDTH met1 >=10  表示找出宽度大于等于 10 的铝。
//————————————————————————————————————
//具体的规则检查
//FLAG CHECK
acute_chk {                        //acute_chk 表示这个错误的名称,随便定义
        @ flag acute yes    //@开头表示注释会在 RVE 的注释窗口里显示
        DRAWN ACUTE //每条检查必须包含一条可以输出错误的命令
}    //一对花括号表示执行一条规则检查
//上面这条规则检查的目的是在版图上直接显示锐角的地方。
offgrid_chk{
         @ flag offgrid yes
         DRAWN OFFGRID
} //直接在版图上显示不在格点上的地方
skew_chk {
          @ flag skew yes
          DRAWN SKEW
}//  直接在版图上显示非 45 度线的地方
//—————————————————————————————————
// well check
GROUP    nwchk    nw_chk? //？ 是一个通配符，这句语句是将所有以 nw_chk 开头的错
                           //误定义成一个名称为 nwchk 的集合。可以通过
                           //DRC SELECT CHECK nwchk  这个语句来控制 DRC 检查
                           //只检查这个集合，也可 UNSELECT 去掉这个集合。


nw_chk1 {
          @ nwell width must >=2.5
          INT nwelli <2.5    ABUT <90 SINGULAR REGION
} // N 阱的宽度检查，后面的 ABUT<90 SINGULAR REGION  是 secong key words  。
  //ABUT<90 一般都要加上，表示有交叉的地方的角度小于 90 度报错
  //SINGULAR 一般也要加上，表示有点碰点或者点碰线的地方都报错
  //REGION  是一个显示控制，表示显示错误时显示范围

nw_chk2{
         @ nwell of same potential space must >=1.4
         EXT nwelli < 1.4    ABUT<90    SINGULAR REGION    SQUARE    CONNECT
} //  同电位的阱间距必须不能小于 1.4  。SQUARE  是输出结果控制，上面有详细说明
 //CONNECT  是连接控制，表示凡是通过铝或其它连接层有连接关系的阱。
```

```
nw_chk3{
        @nwell of different potential space must >=4
        EXT   nwelli <4   ABUT<90 SINGULAR REGION   NOT CONNECT
} //  不同电位的阱间距不能小于 4。

nw_chk4{
        @nwell overlap nsub   >=0.4
        ENC allnsub nwell <0.4 ABUT<90 OUTSIDE ALSO SINGULAR REGION
}//  阱包 nsub 不能小于 0.4, OUTSIDE ALSO  也是 second key words,表示 nsub 在
 // nwell  外也报错。
nw_chk5{
        @ show bad nwell connect two different net
    stamp_nwell = STAMP nwell BY allnsub
    nwell NOT stamp_nwell
}//STAMP 命令来定义 nwell  连接性，并且只能有一个连接，当某个 nwell 的 nsub 有两
//个或以上的不同线名时,这个 nwell 不会被选成 stamp_nwell。没有 nsub 的阱也不会被
//选出来。

//－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－
//关于有源区及多晶硅，铝等层次的检查可参考 nwell 的设置。
//contact check
cont_chk1{
        @min&max contact size 0.5×0.5
        NOT RECTANGLE cont  ＝＝0.5 BY = =0.5 ORTHOGONAL ONLY
}//表示 contact 只能这么大，并且每条边必须都平行与 X 或 Y 轴。
//密度检查
 den_chk1{
        @ min met1 density is 30%
        DENSITY met1 < 0.3 PRINT den_report_m1.log
}//当铝密度小于 30％时，输出 den_report_m1.log 文件，要注意查看。这个文件里有具
//体的铝密度。
```

（6）上面只是一个简单的检查文件，从上面的内容可大致了解一下 Calibre 的检查规则。还有许多其它的检查，如 Antenna（天线效应）、衬底密度等的检查可以参考 Calibre 手册。

2．4 用 commond line 来运行 Calibre DRC 检查

（1） 先必须有一个完整的规则检查文件，必须包含运行设置、层次定义、层次运算、规则检查等几部分。

（2） 在 UNIX 的命令窗口里输入 calibre –hier –drc rulefile 。

（3） 运行完后可在 Cadence 的版图窗口里的 Calibre 菜单点出 start RVE ，然后就可进行 DRC 错误的修改了。

（4） 也可输入 calibre –gui 调出 calibre 图形界面，具体的运行方式类似 Cadence 环境下的模式。

2．5 查看结果文件和改错

1、 drc 检查运行完毕后， 首先看drc_err.sum 文件，看有无错误。下面是一个例子：
前面是本次运行的一些信息：

CALIBRE:: DRC-F SUMMARY REPORT
Execution Date/Time: Fri Jan 2 20:10:46 2004
Calibre Version: v9.1_9.3 Fri Dec 13 15:05:27 PST 2002
Rule File Pathname: drc.rule
Rule File Title:
Layout System: GDS
Layout Path(s): test_nand3.gds
Layout Primary Cell: test_nand3
Current Directory: /export/home/project/cpu863/LVStest/calibre/drc/test
User Name: cpu863
Maximum Results/RuleCheck: 1000
Maximum Result Vertices: 4096
DRC Results Database: drc.out (ASCII)
Layout Depth: ALL
Text Depth: PRIMARY
Summary Report File: drc_err.sum (REPLACE)
Geometry Flagging: ACUTE = YES SKEW = YES OFFGRID = NO
NONSIMPLE POLYGON = YES NONSIMPLE PATH = NO
CheckText Mapping: COMMENT TEXT + RULE FILE INFORMATION
Layers: MEMORY-BASED
Keep Empty Checks: YES

--- RUNTIME WARNINGS ——
There is no data for layout net name ?vcc?.

中部开始有关RuleCheck Results 的统计，如下：

--- RULECHECK RESULTS STATISTICS

RULECHECK NW_1 ..... TOTAL Result Count = 0
RULECHECK NW_2a ..... TOTAL Result Count = 0
RULECHECK NW_2b ..... TOTAL Result Count = 0
RULECHECK NW_3 ..... TOTAL Result Count = 0
RULECHECK NR_1 ..... TOTAL Result Count = 0
RULECHECK NR_2 ..... TOTAL Result Count = 0
RULECHECK NR_3 ..... TOTAL Result Count = 0
RULECHECK NR_4 ..... TOTAL Result Count = 0
RULECHECK NR_5 ..... TOTAL Result Count = 0
RULECHECK NR_6 ..... TOTAL Result Count = 0
RULECHECK NR_7 ..... TOTAL Result Count = 0
RULECHECK NR_8 ..... TOTAL Result Count = 0

如RULECHECK NR_8 ..... TOTAL Result Count = 0 表示NR_8 这条规则检查的结果是0 个错误，具体NR_8 规则的含义要看DRC Runset File 中的定义，

检查有错的如：

RULECHECK CT_9 ..... TOTAL Result Count = 0
RULECHECK CT_10 ..... TOTAL Result Count = 0
RULECHECK CT_11 ..... TOTAL Result Count = 0
RULECHECK CT_12 ..... TOTAL Result Count = 0
RULECHECK M1_1 ..... TOTAL Result Count = 0
RULECHECK M1_2 ..... TOTAL Result Count = 1
RULECHECK M1_3&4 ..... TOTAL Result Count = 0
RULECHECK M1_5 ..... TOTAL Result Count = 0
RULECHECK M1_6 ..... TOTAL Result Count = 3
RULECHECK M2_1 ..... TOTAL Result Count = 0
RULECHECK M2_2 ..... TOTAL Result Count = 0
RULECHECK PD_M1 ..... TOTAL Result Count = 0
RULECHECK PD_M2 ..... TOTAL Result Count = 1
RULECHECK PD_M3 ..... TOTAL Result Count = 1
RULECHECK PD_M4 ..... TOTAL Result Count = 1
RULECHECK PD_M5 ..... TOTAL Result Count = 1
RULECHECK PD_M6 ..... TOTAL Result Count = 1
RULECHECK Convention_FLT_NW .... TOTAL Result Count = 0

分别是1，3，1，1，1，1 个错误，最后是错误统计：

--- SUMMARY

TOTAL CPU Time: 0
TOTAL REAL Time: 1
TOTAL Original Layer Geometries: 73
TOTAL DRC RuleChecks Executed: 205
TOTAL DRC Results Generated: 9

可见一共检查出9 个DRC 错误。如果没有错，则最后的TOTAL DRC Results Generated 为0。

还可以查看DRC 检查结果的数据库“drc_err”， 如下：

test_nand3 1000
NW_1
0 0 2 Jan 2 20:10:47 2004
Rule File Pathname: drc.rule
Minimum width of an NW region is 0.86um
NW_2a
0 0 3 Jan 2 20:10:47 2004
Rule File Pathname: drc.rule
Minimum space between two NW regions with the same potential is 0.60um
Merge if space is less than 0.6um
NW_2b
0 0 2 Jan 2 20:10:47 2004
Rule File Pathname: drc.rule
Minimum space between two NW with different potential is 1.40um

NW_3
……
Rule File Pathname: drc.rule
NW without N+ pick up
Convention_BPMO
0 0 2 Jan 2 20:10:48 2004
Rule File Pathname: drc.rule
Pmos in PW
Convention_BAD_IMP
0 0 2 Jan 2 20:10:48 2004
Rule File Pathname: drc.rule
AA area without any implant
__RVE_ERROR_TAG2__
0 0 14 Jan 11 10:10:57 2004
M1_2 151 1
1
M1_6 154 1
000
PD_M2 197 1
0
PD_M3 198 1
0
PD_M4 199 1
0
PD_M5 200 1
0
PD_M6 201 1
0

也包含了检查的错误信息，该数据库主要是被后面的RVE 来调用的。然后要根据错误去版图中相应的位置改，Calibre 提供了良好的**RVE**（**R**esult **V**iew **E**nviroment）界面，它能直接调用DRC 或者LVS 检查后的结果数据库，图形化很直观地显示错误所在，并且可以调用版图工具如Virtuoso，直接在Virtuoso 中快速定位错误位置。

2、用 **RVE** 查看结果和改错

RVE（**R**esult **V**iew **E**nviroment）是Calibre 自带的看验证结果的集成工具。启动方式为：

calibre –rve database

database 为DRC 或LVS 检查结果的数据库，这里是drc_err ：

calibre -rve drc_err

激活后界面如图1

可见左边显示DRC 错误种类和数目，右边显示坐标位置，下面是对该Rule 的解释。清晰的界面，方便的操作能帮助用户快速找到错误和修改。RVE 能调用很多版图工

具，把错误直接反映在版图位置上，见菜单Setup ——> Layout，这里可以设置调用的版图工具。具体配置方法见在线帮助中Calibre 与其他工具的接口。在Cadence的Virtuoso 中集成了Calibre 以后，Virtuoso 的菜单中会多出Calibre 的菜单，如：即可以直接从Virtuoso 中调用图形化的DRC, LVS 和RVE，显得十分方便。点击Start RVE ， 选择数据库名称，同样可以得到同图1 一样的界面，这时RVE 已经和版图工具集成在一起，可以在RVE 中点击一些错误坐标，Virtuoso 中立刻会显示错误的位置，如图3：

这里点击M1 小于最小面积的错误中的第1 个错误点：右边坐标红色下划线的(-1.96 , 3.97)，在版图工具中会立刻高亮显示位置，如图4：

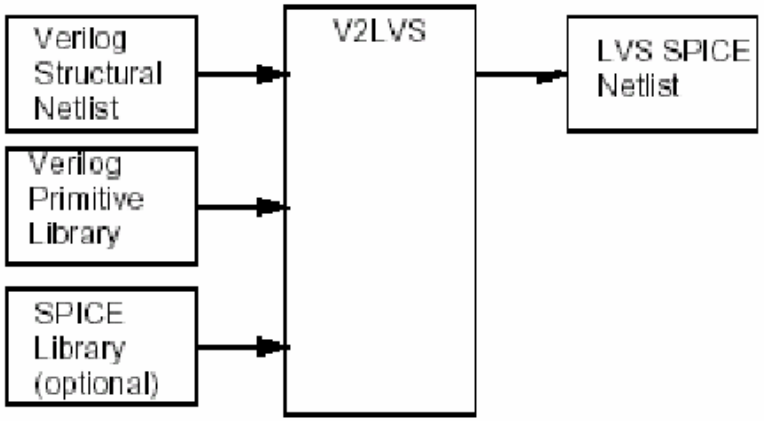
因此改错起来十分方便。可以在RVE 菜单View ——> By Check , RVE 左边会显示出Check 每条rule 的结果，见图5：

绿色的表示检查该rule 无误。而且利用File 菜单中可以方便地打开DRC 结果报告、DRC Runset 等，如图6：

第三章 Calibre LVS

3．1 数据准备

需要的数据为版图数据、电路图数据和**runset** 文件。Calibre 把电路图网表的部分叫SOURCE。SOURCE 部分要求的网表格式为标准spice 格式或者Calibre 自身的一种类似spice 的格式。Calibre 有一种把verilog 转为自身类spice 格式的功能，叫v2lvs，下面先介绍v2lvs：
v2lvs 能够把verilog 网表和相对应的spice 库、verilog 子库转成Calibre LVS 用到的电路图SOURCE 网表，其功能如图：



转换的原理是verilog 网表根据verilog 子库对元件端口的定义，去spice 子库找同样名称和端口的元件，然后进行网表和格式上的替换。因此v2lvs 转类spice 网表时，需要verilog 网表、spice 库、verilog 子库描述（可选）,其命令格式如下：

v2lvs –v verilog_file –o spice_like_file [-l verilog_lib_file] [-lsp spice_lib_file] [-lsr spice_lib_file] [-s spice_lib_file] [-s0 groundnet] [-s1 powernet] [-sk] [-i]

-v 后面接verilog 文件名称；-o 为输出类spice 格式文件；-l 是verilog 子库描述；-lsp 接spice 库网表，p 是pin 模式，即不允许有数组类的verilog 端口(比如PA[3]、PA[2]、PA[1]、PA[0])出现；-lsr 与-lsp 意义同，不过pin 是range 模式，即可以接受verilog 的数组端口；与-lsp，-lsr 不同的是，-s 是只是让转出文件在前面INCLUDE 这些spice 子库，而不会读它们；-s0 , -s1 为对verilog 中1'b0, 1'b1 的电源网络取代；-sk 指允许多组复合电源，不仅仅是一对全局电源VDD,VSS；-i 指输出文件采用spice 通用的pin 格式，没有$引导，便于仿真。常见的例子如下：

v2lvs –v top_design.hv –o top_design.sp –s0 VSS –s1 VDD –sk –l pll_risc.v -l cache_core.v –l pad.v –l std.v -lsr pll_risc.sp –lsr cache_try_new.sp –lsp std.sp -lsp pad.sp -s pll_risc.sp –s cache_try_new.sp –s std.sp –s pad.sp

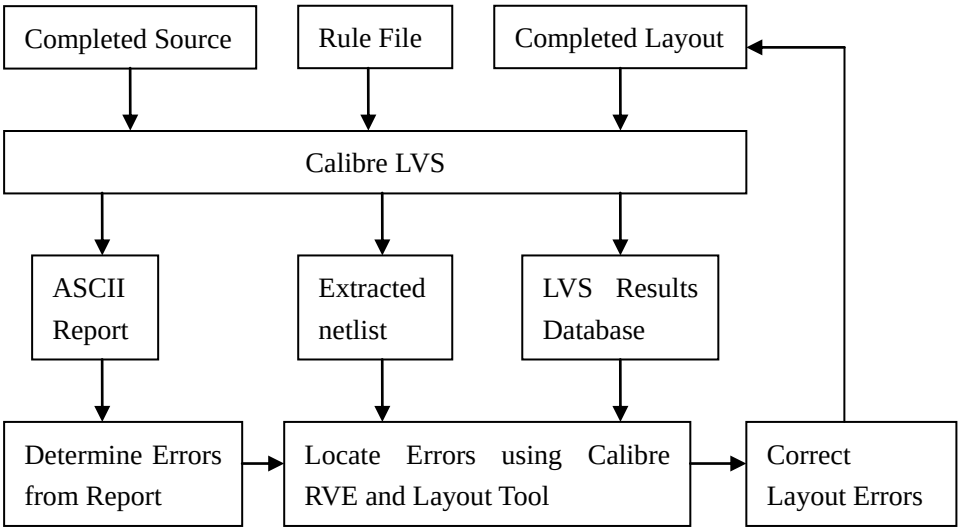
根据几个子模块的情况，转出整个芯片的LVS 网表。

实际上，verilog 子库并不是都需要的，v2lvs 在转spice 网表的时候，是根据verilog 子库中元件端口和spice 子库中的元件端口名称对应关系来转的。如果在verilog 网表中没有数组端口,则该元件端口名称在spice 子库网表中元件的端口名称中也是唯一的标识，因此不会转错。但是对于数组端口（如PA[31：0]），如果没有verilog 子库对模块的端口定义，v2lvs 按照缺省的从高位到低位的方式转出端口对应关系，因此会出错。因此对于有数组端口的**verilog** 网表，一定要求有**verilog**子库网表。上面的例子可以简化为：

v2lvs –v top_design.hv –o top_design.sp –s0 VSS –s1 VDD –sk -l pll_risc.v –l cache_core.v -lsr pll_ric.sp -lsr cache_try_new.sp -lsp std.sp –lsp pad.sp -s pll_risc.sp –s cache_try_new.sp –s std.sp –s pad.sp

3．2 LVS 流程

从下面的流程中可看出，Calibre LVS 都先把版图提取出 SPICE 格式的网表来，实际上 Calibre LVS 比对的是两个 SPICE 网表。



3．3 一个简单的 LVS Runset 文件

```
// 注释
LAYOUT PATH  "/home/*"    //版图数据的路径及名称
LAYOUT PRIMARY  "*" //版图的顶单元名
LAYOUT SYSTEM GDSII //版图数据的格式
//-------------------------------------------------------------
SOURCE PATH  "/home/*" //逻辑网表的路径及名称
SOURCE PRIMARY  "*" //顶层逻辑名
SOURCE SYSTEM SPICE  //逻辑网表的格式
//-------------------------------------------------------------
LVS REPORT  "/home/*" //LVS Report 的路径及名称
LVS REPORT  OPTION  S //显示软连接的冲突
PRECISION  1000
RESOLUTION  10  //格点为 0.01
UNIT CAPACITANCE    FF
UNIT RESISTANCE     OHM
UNIT LENGTH         U  //定义文件里电容、电阻、长度的单位
//-------------------------------------------------------------
LVS POWER NAME    "?VDD?" "?vdd?" "?VCC?" "?vcc?" //定义电源 ？为通配符
LVS GROUND NAME  "?GND?" "?gnd?" "?VSS?" "?vss?"//定义地
TEXT DEPTH PRIMARY //  定义只认顶层的 TEXT，用 ALL 表示认识所有的 TEXT
VIRTUAL CONNECT COLON YES//定义允许用名字的虚拟连接
 //-------------------------------------------------------------
LVS ABORT ON SUPPLY ERROR YES //电源地短路就中断 LVS,要检查短路才设为 NO
LVS ISOLATE SHORTS  NO //发现电源地短路时改为 YES，因为它需要很长时间
LVS IGNORE PORTS    NO  //不忽略 PORTS
LVS CHECK PORT NAMES  YES //比 PORTS 的名字
LVS RECOGNIZE GATES     ALL//  同上面图形界面里 Gates 的设置
LVS ALL CAPACITOR PINS SWAPPABLE  YES //允许电容两端互换
//-------------------------------------------------------------------
LVS FILTER UNUSED MOS  YES //忽略版图里不用的 MOS 管，下面有详细控制
LVS FILTER UNUSED RESISTORS YES //忽略版图里不用的电阻，下面有详细控制
LVS FILTER UNUSEDC CAPACITORS YES //忽略版图里不用的电容，下面有详细控制
LVS FILTER UNUSED OPTION AB RC RE RG YC O
//定义哪些器件是不用的器件：AB 表示去掉 G S D 三端连一起的 MOS 管，RC 表示去
//两端连一起的电阻，RE 表示去掉两端连一起的电容，RG 表示去掉两端连一起的二极
//管，YC 表示去掉三端连一起的三极管，O 是过滤的重复设置，一般都加上。
LVS REDUCE PARALLEL BIPOLAR YES //将并联的三极管当成一个
LVS REDUCE PARALLEL MOS     YES //将并联的 MOS 管当成一个
LVS REDUCE PARALLEL DIODES YES //将并联的二极管当成一个
LVS REDUCE PARALLEL CAPACITORS YES //将并联的电容当成一个
LVS REDUCE PARALLEL RESISTORS YES //将并联的电阻当成一个
LVS REDUCE SERIES RESISTORS YES //将串联的电阻当成一个
LVS REDUCE SERIES CAPACITORS YES //将串联的电容当成一个
//-------------------------------------------------------------------------------
```

```
//输入层次定义
LAYER   nwelli 1
LAYER    ndiffi 2
LAYER    pdiffi 3
        .
        .
        .
bulk =    EXTENT //定义大衬底，EXTENT 表示数据的最外框
TEXT LAYER 60 ATTACH 60 met1
PORT LAYER TEXT 60
TEXT LAYER 70 ATTACH 70 met2
PORT LAYER TEXT 70    //定义 TEXT 的连接
LAYOUT TOP LAYER padi met3i met2i via2i via1i //   将这些数据当成顶层
//-------------------------------------------------------------------------------------
//层次运算，同 DRC
ndiffx = rlocosi    AND    nimpi
            .
            .
            .
//-------------------------------------------------------------------------------------
//定义连接，同 DRC
CONNECT met2 met1 BY via1
            .
            .
            .
//-------------------------------------------------------------------------------------
//器件定义
DEVICE MN(N) ngate tpoly tnsd tnsd bulk [0.5] //定义 NMOS 管，拐角因子为 0.5
TRACE PROPERTY MN(N) L L 0.1 //比对沟道长度  允许 10%的误差
TRACE PROPERTY MN(N) W W 0.1//比对沟道宽度  允许 10%的误差
DEVICE MP(P) tpgate tpoly tpsd tpsd tnwell [0.5] //定义 PMOS 管，拐角因子为 0.5
TRACE PROPERTY MP(P) L L 0.1 //比对沟道长度  允许 10%的误差
TRACE PROPERTY MP(P) W W 0.1 //比对沟道宽度  允许 10%的误差
DEVICE R(RW) bnwell tnsd tnsd [800] //定义电阻，方块阻值为 800OHM
TRACE PROPERTY R(RW) R R 0.3 //比对电阻值，允许 30％的误差
DEVICE C(CL) pccap cpo2 tpoly [0.72 0] //定义电容，每平方电容值 0.72FF，0 表示不计
                                        //算周长效应的电容
TRACE PROPERTY C(CL) C C 0.3 //比对电容值，允许 30％的误差
DEVICE Q(PL) TRImk coll base emit //定义三极管
DEVICE D(DN) ndiomk bulk tndio //  定义二极管
TRACE PROPERTY D(DN) A A 0.3 //比对二极管的面积，允许 30％的误差
//-------------------------------------------------------------------
//ERC CHECK  部分
LVS SOFTCHK tnwell CONTACT
```

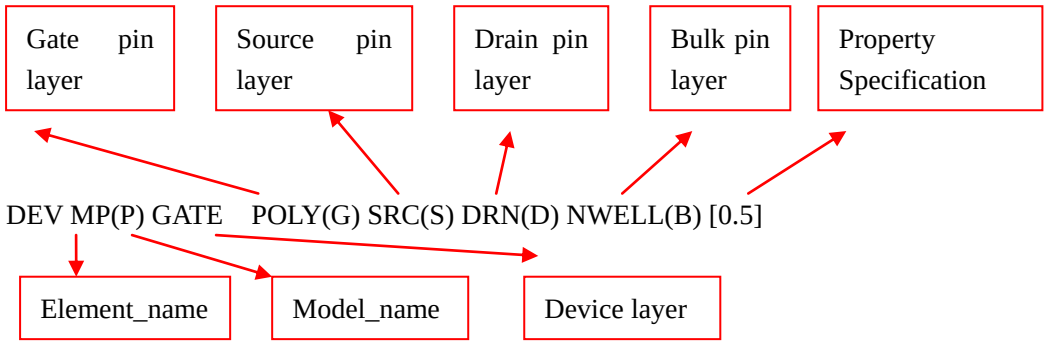
```
LVS SOFTCHK bulk    CONTACT
ERC PATHCHK GROUND&& !POWER
ERC PATHCHK POWER&& !GROUND
ERC PATHCHK !POWER&&!GROUND
ERC PATHCHK !LABELED
psub_to_power {
                    NET psub “?VDD?” “?vdd?” “?VCC?” “?vcc?”
}
nsub_to_ground {
                    NET nsub “?VSS?” “?vss?” “?GND?” “?gnd?”
}
```

3．4　Runset 文件里器件的定义

器件定义的一般格式

DEVICE　element_name [(model_name)] device_layer {pin_layer} [property_spectification]

Example:



Element name	Definition	Pin names	Default Properties for Tracing	Parameters for Property Specification
MN MP MD ME	MOS Transistor	G(gate) 1st pin layer S(source) 2nd pin layer D(drain) 3rd pin layer B(bulk) 4th pin layer is optional	Width Length	Effective_width_factor (weffect)
D	Diode	POS(+pin) 1st pin layer NEG(-pin)2nd pin layer SUB(substrate)3rd pin layer is optional	Area Perimeter	

C	Capacitor	POS 1st pin layer NEG 2nd pin layer SUB 3rd pin layer is optional	Capacitance	Area_cap Perim_cap
R	Resistor	POS 1st pin layer NEG 2nd pin layer SUB 3rd pin layer is optional	Resistance	resistivity
Q	Bipolar Transistor	C(coll) 1st pin layer B(base) 2nd pin layer E(emit) 3rd pin layer SUB 4th pin layer is optional	NONE	

3．5 用 commond line 来运行 Calibre LVS 检查

（1）先必须有一个完整的规则检查文件，必须包含运行设置、层次定义、层次运算、器件定义等几部分。

（2）建一个批处理命令比如叫 run_lvs，内容为

```
＃！ /bin/csh –f
\rm –r /home/*…/svdb
calibre –lvs –hier –spice /home/*…/svdb/topcell.sp –hcell  hcell_file rule_file |tee lvs.log
//假如没有 hcell_file
calibre –lvs –hier –spice /home/*…/svdb/topcell.sp –auto rule_file |tee lvs.log
```

（3）前面 rm 的目的是每次清空 LVS Database 目录，命令中 topcell 为版图的顶层单元名。

（5）然后每次运行这个批处理命令就可以了，在 Unix 命令行下输入 ./run_lvs 命令就可。运行完后可以到存放结果文件的目录里看 Report File，还可以在版图里用 RVE 调 LVS Database 进行 Debug

（6）hcell_file 的格式如下

// 是注释。

```
//Layout cell Name        Source cell Name
   ABC                        DEF
   ABC                        GHI
   UVW                        XYZ
   RST                        XYZ
```

可见单元名可以 1 VS N 或者 N VS 1 ，但是不要出现 M VS N 的情况。

3．6 查看结果文件及改错

（1）打开 LVS 结果文件，下面是个实例：

OVERALL COMPARISON RESULTS

```
#      #      ##############
 #   #        #             #
  #            #   INCORRECT   #
 #   #        #             #
#      #      ##############
```

Error: Different numbers of nets.
Error: Different numbers of instances.
Error: Connectivity errors.
Error: Instances of different types or subtypes were matched.
Error: Property errors.
Warning: Unbalanced smashed mosfets were matched.
Warning: Ambiguity points were found and resolved arbitrarily.

****

*** CELL SUMMARY ***

****

Result	Layout	Source
INCORRECT	TOPCELL	TOPCELL

可见总的比较结果是不正确，错误有5 种： 不同网络、不同元件个数、连接错误、不同元件类型、属性错误，CELL SUMMARY 里面有Layout 和Source 的TOPCELL 不匹配。然后直接翻页到文件的后面，看到**INFORMATION AND WARNINGS** 栏：

****

INFORMATION AND WARNINGS

****

这里列出了匹配的统计情况（同dracula lvspr.lvs 的最后），可以看见SOURCE 和LAYOUT 匹配了多少，各有多少没有匹配，错在哪种单元上面等，这里可以看出共有10 个layout 单元和16 个source 单元没有匹配。下面是管子等删减的情况统计：

o Statistics:
310940 layout mos transistors were reduced to 94596. 24286 connecting nets were deleted.
171921 mos transistors were deleted by parallel reduction.
16 mos transistors and 16 connecting nets were deleted by series reduction.
44407 mos transistors and 24270 connecting nets were deleted by split-gate reduction.
106383 source mos transistors were reduced to 28181. 22460 connecting nets were

deleted.
36983 mos transistors were deleted by parallel reduction.
16 mos transistors and 16 connecting nets were deleted by series reduction.
41203 mos transistors and 22444 connecting nets were deleted by split-gate reduction.
20 series/parallel layout resistors were reduced to 6. 8 connecting nets were deleted.
141 unused layout mos transistors were deleted.
141 unused source mos transistors were deleted.
2 unused layout resistors were deleted.
47 nets were matched arbitrarily.

下面是顶层端口对应情况,这个很重要：

Initial Correspondence Points:
Ports: vdd VDD33 SAVDD VSS VSSD SAVSS EB_RDVAL EB_WDRDY EB_RBERR EB_WBERR SI_NMI SI_INT[5] SI_INT[4] SI_INT[3] SI_INT[2] SI_INT[1] SI_INT[0] SI_ENDIAN SI_COLDRESET EB_ARDY SI_RESET EJ_DINTSUP EJ_DINT CACHE_CE TRST TMS TCK TDI SI_ERL SI_EXL SI_RP SI_CLKOUT SI_SLEEP SI_TIMERINT EB_AVALID EXT_CLK EXT_CLKEN EB_INSTR EB_WRITE EB_BURST EB_BFIRST EB_BLAST EJ_DEBUGM EB_BE[3] EB_BE[2] EB_BE[1] EB_BE[0] FREF PLL_TEST SI_PLLCLK

如果Ports 部分没有出现顶层模块的所有端口， 则肯定会导致整个比较的失败，因此如果端口方面信息错误的话， 应该去查一下版图抽取的时候是否出了问题，可以去看TOPCELL.sp 和TOPCELL.rep.ext。注意： Ports 报告的数目和LVS Report MAXIMUM number 的设置数据有关，最好使number 大于顶层的端口数目。下面是具体详细的连接信息。如果发现端口基本正确，可以直接到文件的前面看详细的 Error 信息。**CELL SUMMARY** 下面是**LVS PARAMETERS** 部分，即回放LVS 比较的所有选项设置：

看TOPCELL.rep 的方法是：先看**OVERALL COMPARISON RESULTS**，看总体比较是否正确，如果不对，看错误的类型；然后去文件后面看**INFORMATION AND WARNINGS** ， 看具体元器件匹配的情况， 尤其是下面的**Initial Correspondence Points Ports** 中顶层端口匹配的情况，如果端口没有匹配好，去查版图抽取的情况；如果出现很多大量的管子不匹配，请检查电源网络是否有开路，短路现象，同时可以设置LVS Recongnize gates 和LVS Reduce split gates 为yes 再试一次（即允许pin 交换和删减复杂门）；最后可以回到文件开头看**INITIAL NUMBERS OF OBJECTS** 和**NUMBERS OF OBJECTS AFTER TRANSFORMATION** ， 以及具体的**INCORRECT NETS** ， **INCORRECT INSTANCES**，**PROPERTY ERRORS** 信息，在版图和电路图上找到相对应的地方进行检查。

注意：对于短路现象，则Layout 中的网络数目必然少于Source 中的，并且出现Layout 中几个网络对应与Source 中的一个网络的信息；对于开路，Layout 中的网络数目必然大于Source 中的，且Source 中的一个网络可以对应与Layout 中的几个网络。

3． 7 用 RVE 看结果及改错

与DRC 一样在运行目录下激活RVE 或者在Virtuoso 中Start RVE ：

calibre -rve svdb

svdb 是LVS Runset 中MASK SVDB DIRECTORY "svdb" QUERY 中所确定的数据库名称，LVS 所有信息（包括版图抽取）都存放在该数据库。

当用 RVE 窗口来 Debug 时，在版图的 Calibre 菜单下的 Setup 的 RVE 的设置里选中 Edit-in-place while highlighting，在 Layout cells 的命令行里输入版图库的名称，在 Schematic cells 的命令行里输入逻辑库的名称。

RVE 激活后，打开窗口如图 7

可以看见被Query 的Cell 叫cache_core，即本设计的TOPCELL。RVE 的左边一栏是：输入文件：Runset 文件、SOURCE 文件；输出文件：Layout 网表、抽取报告、LVS 比较报告、ERC 电气检查结果。这些都可以用鼠标单击即可打开，如下面的 Runset 文件：

同样可以点击打开Source Netlist：

可见SOURCE 的网表的层次十分清楚，这都很有利于后面的对应到版图和改错。打开LAYOUT 网表：

Layout 的网表层次关系也十分清楚，这些都有利于理清版图的层次。抽取版图的信息报告（TOPCELL.rep.ext）如下：

LVS 比较的结果报告（TOPCELL.rep）如下：

RVE 的右边一栏（见图7）是错误的列表，每种错误类型（Discrepancy）和具体错误的个数；下面一栏是具体某个错误的详细信息，比如座标位置。错误的详细信息已经高亮度显示（黄色,蓝色，绿色），右键点击左边Layout Name 栏的错误点，选

Zoom to Point，如图：

在Virtuoso 中会立刻高亮显示该处的位置。这样每个错误都可以直接定位到版图上，改起来很方便。同时还可以选上图的Highlight Closeset，可以高亮附近的网络和器件。更为巧妙的是，图7 下面Source Name 栏相对应的错误点也可以直接在版图工具中显示，右键点Source Name 对应的错误点，选Highlight Net，如图15：

Calibre 会打开新的Virtuoso 窗口，如图16：

然后在新打开的窗口高亮该网络：

这样很方便Discrepancy 中Layout 与Source 具体位置的对比，便于及时找到错误。对于图10 中Layout Netlist 中的每一个网络名称和元件，都可以直接在Layout Netlist窗口中单击该名称，在Virtuoso 中会立刻显示，就像超链接一样，见图18：

图18 询问用户是否现在显示该网络或者元件，选Yes , Virtuoso 中会高亮该网络或器件。

这也很容易理解，因为Layout Netlist 本来就是Calibre 根据Runset 规定的层次抽取出来的，因此每个网络或者元件，数据库中都有记录。对于图9 的Source Netlist ，同样可以用类似的方法点击网络或元件（只对顶层模块），来在版图工具中显示。因此利用RVE 读入LVS 的结果数据库把电路图网表，版图网表，LVS 比较结果紧密地联系起来，集成在非常方便的图形工具中，用超链接的方式快速找到每个节点，每个器件与版图的对应关系，又凭借着电路图网表窗口和版图网表窗口良好的层次化结构，因此能很快的找到错误的所在。这也是Calibre RVE 最出色的方面之一。