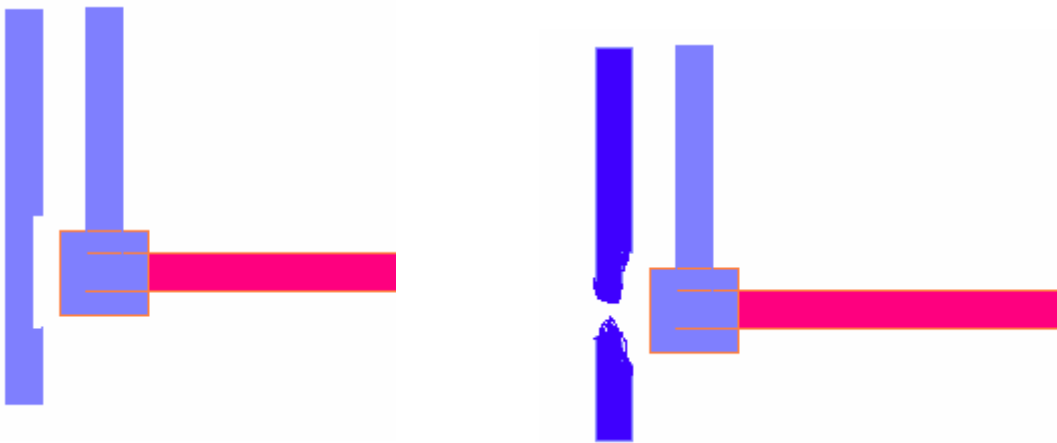


# DRC/LVS

## 8.1 DRC 概述

在第七章中讲到了怎样把前端综合出来的 netlist 实现成版图 (layout)、怎样用静态时序分析证明所实现的版图在时序上满足要求, 现在还有两个任务没有完成, 一是要证明这个已经实现的版图可以生产出来, 也就是说这个版图符合流片厂家的要求, 这一步称为 DRC, 另一个任务是证明这个版图就是想要的那一个, 也就是说这个版图实现的功能与网表描述的相一致, 这一步称为 LVS。DRC 和 LVS 是比较复杂的, 本章只能简要介绍其基本概念, 进一步的介绍请查阅各 DRC/LVS 工具的文档。

现在先介绍 DRC (Design Rule Check), 做 DRC 的目的是保证版图满足流片厂家的设计规则, 因为不是任何版图都能制造出来, 只有满足厂家设计规则的版图才有可能成功造出来。例如, 如果厂家的设计规则中有一条 “金属 1 的最小宽度是 0.5um”, 那么假如版图中有地方金属 1 的宽度为 0.4um, 用这个版图去流片, 流出的片子有可能在这个地方断路。



如图 T8.1 所示。因此, 需要有 DRC 工具来检查版图是否符合这些几何规则。

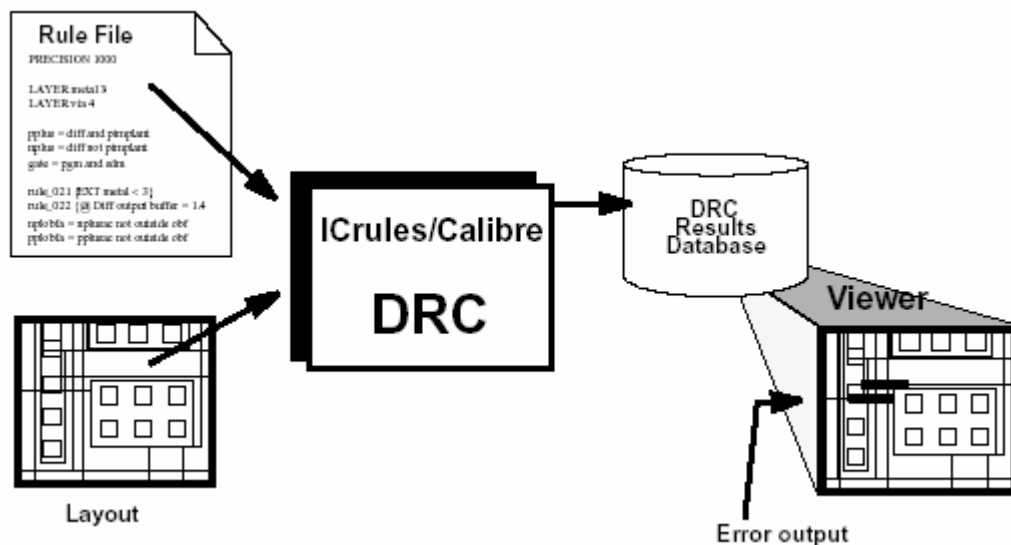
T8.1 左边为版图, 右边为实际制出的形状

为了保证版图能正确制出, 流片厂家会根据工艺定义很多的设计规则, 只有版图满足厂家的所有的设计规则, 才可能被正确的制出。一般说来, 设计规则有很多, 例如 最小间距、最小孔径等。

做 DRC 的工具具有 cadence 的 dracula、avanti 的 hercules、mentor 的 calibre 等。本章将结合 mentor 的 calibre 来介绍 DRC。下面介绍的所有概念和命令都是基于 mentor 的 calibre 的。其他的 DRC/LVS 工具会有不同的概念和命令。

关于 DRC、LVS, 有一个概念要明确, 就是无论是手工画出的版图或者 APR 工具自动生成的版图, 如果不经 DRC&LVS 验证, 那么这个版图极有可能是错的, 也就是说, 不经验证的版图几乎一定是错的, 尤其是当 chip 的规模越来越大时。DRC、LVS 工具就是辅助 designer 发现并改正错误。只有经过严格验证的版图才有可能是 designer 想要的正确的版图。

## 8.2 DRC Rule File



T8.2 DRC flow

如图 T8.2 所示，DRC 的输入有两项，一个是 layout，就是手工或 APR 生成的版图，一般是 GDSII 格式。另一个是 Rule File，Rule File 告诉 DRC 工具怎样做 DRC，这个文件十分重要，一般，由流片厂家提供，或者由 designer 根据流片厂家提供的版图几何规范自己写。Calibre 读入 GDS（版图）和 Rule File，进行处理，输出结果，输出结果是 calibre 自己定义的格式，designer 可以通过一个 Viewer 来看，一般用 cadence 的 virtuoso 来分析输出结果和修改版图。

做 DRC 时 designer 最先应该知道的是 Rule File，如果 Rule File 是厂家提供的，那么 designer 应该知道 Rule File 的内容，因为将来要根据它来修改查出有错的版图，如果要 designer 根据厂家提供的版图几何规范来自己写 Rule File，那么 designer 就更要了解 Rule File 的内容、语法。下面是一个简单的 Rule File，右边是对它的解释。

```

layout system gdsii           # 指明输入的版图格式是 GDSII，calibre 还可支持其他格式
layout path "/layout/basic_drc.gds" # 输入版图的路径
layout primary basic_drc      # 要检查的版图的 top cell 名字
drc results database "/training/drc_results_db" # drc 的输出，将来可以用
                                                # virtuoso 分析

variable metal_width 3        # 定义一个变量

layer substrate 0             # 本行和下面相似的行定义了输入版图中每一层的名字
layer pwell 1                 # 本行把 GDS 的 layer 1 定义为 pwell，接下来 pwell 都指输入
                              # 入的 GDS 的 layer1。

layer oxide 2
layer res 3
layer poly 4
layer nplus 5
layer pplus 6
layer contact 7
layer metal1 8
  
```

```
layer via      9
layer metal2   10
```

```
ngate = poly AND nplus      # layer 运算，由原始的 poly 和 nplus 层推出 ngate 层
...
//
// Module 2 - DRC Measurement Operations
// Internal Checks
rule_2a { @oxide minimum width is 6      // 进行检查，rule_2a 是检查标号，
      // @oxide minimum width is 6      是解释，报错时会报出这一句，
internal oxide < 6 }      // 这一句是具体动作，internal 是 calibre 的关键字，
      // 这一句检查所有的 oxide 的 internal 小于 6 的情况。
      // 如果 internal oxide < 6 的情况出现，则输出报错结果
      // designer 根据这个结果修改错误。
      // internal 具体介绍见下一节
rule_2b { @oxide minimum overlap of pwell is 3 // 第二项检查
internal oxide pwell < 3 }      // 检查 oxide 与 pwell 的内间距
rule_2c { @ pwell minimum width is 4      // 第三项检查
internal pwell < 4 }      // pwell 的内间距小于 4 就报错
//
// Module 2 - DRC Measurement Operations
// External Checks
//
rule_3a { @res minimum spacing is 2
external res < 2 }      // 检查 res 的外边距
rule_3b { @res minimum spacing to oxide is 2
external res oxide < 2 }
//
// Module 2 - DRC Measurement Operations
// : Enclosure Checks
//
rule_4a { @ res minimum enclosure of oxide is 7
enclosure oxide res < 7 }      // oxide 的内外边的间距，enclosure 具体的介绍见下一节
rule_4b { @ oxide minimum enclosure of res is 6
enclosure res oxide < 6 }
rule_4c { @ oxide minimum enc of res is 6 - res out only
enclosure [res] oxide < 6 }
```

一般说来，Rule File 首先要设好 calibre 运行的环境，包括 GDS 文件的位置、输出文件的位置等。接下来要定义输入的 GDS 版图的层次，例如如果 GDS 的 layer 1 是多晶硅层，则应该定义如下

```
layer poly    1
```

这句话表明所有 GDS 的 layer1 现在都叫 poly（当然也可以叫任何名字）。接下来可以用 poly 来代表 layer1。之所以要这样做是因为在 GDS 内，所有的层都用一个数字来表示，

如 1, 2, 3...GDS 文件并不知道 layer1 是什么, calibre 也无法从 GDS 中知道 layer1 是什么, 只有 designer 来告诉 calibre。

下面是 GDS 文件的一部分转成 ASCII 后的显示:

Begin Cell Definition

Cell Name : VG, View Name : layout

Rectangle - Layer : 7 Data Type : 0 BBOX : (5000,-10500) (16000,-8100)

Rectangle - Layer : 7 Data Type : 0 BBOX : (5000,5900) (16000,8300)

Rectangle - Layer : 1 Data Type : 0 BBOX : (5000,0) (16000,10200)

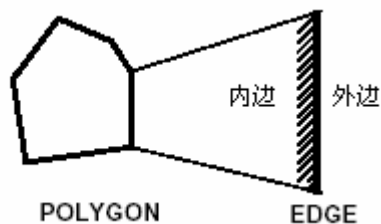
End Cell Definition

从这一段可以看出, GDS 文件只是描述版图的几何形状, 所有的 layer 都是一个整数。用 GDS 描述的版图不能表示层的具体信息, 只能由 designer 来定义。

定义了层, 接下来就是进行层的运算(与, 或, 非...), 然后判断版图是否符合设计规则。具体概念见下一节。

### 8.3 DRC 规则

Calibre 是一个基于边(edge)的 DRC/LVS 工具, 所有的运算都是基于“边”来进行的。边有内边和外边之分。



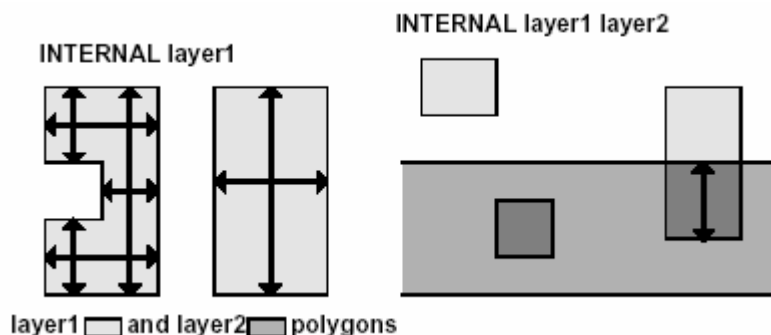
T8.3 edge

如图 T8.3 所示, 任一个 polygon 的边都分为两部分, polygon 内部那一面的称为内边 (interior side edge), 外部的称为外边 (exterior side edge)。理解这个定义对于理解 calibre 复杂的指令是十分有用的。

下面介绍几个 calibre 基本的例子。通过这些例子有助于理解 calibre 的工作原理。

#### 8.3.1 Internal

Internal 指令一般用于检查 polygon 内间距, 可以用来检查同一 layer 的 polygon 的内间距、或检查两个不同 layer 的 polygon 之间的内间距。如图 T8.4 所示:



T8.4 Internal operation

理解 Internal 指令的关键是 Internal 指令是 polygon 的内边 (interior edge) 的相对关系, 只有 interior edge 之间才能检查。T8.4 的左边是在同一个 polygon 的 internal 检查, 请注意左边凹进去的相对两边不作检查, 这是因为这两边是 exterior edge。同一 layer 的 polygon 的

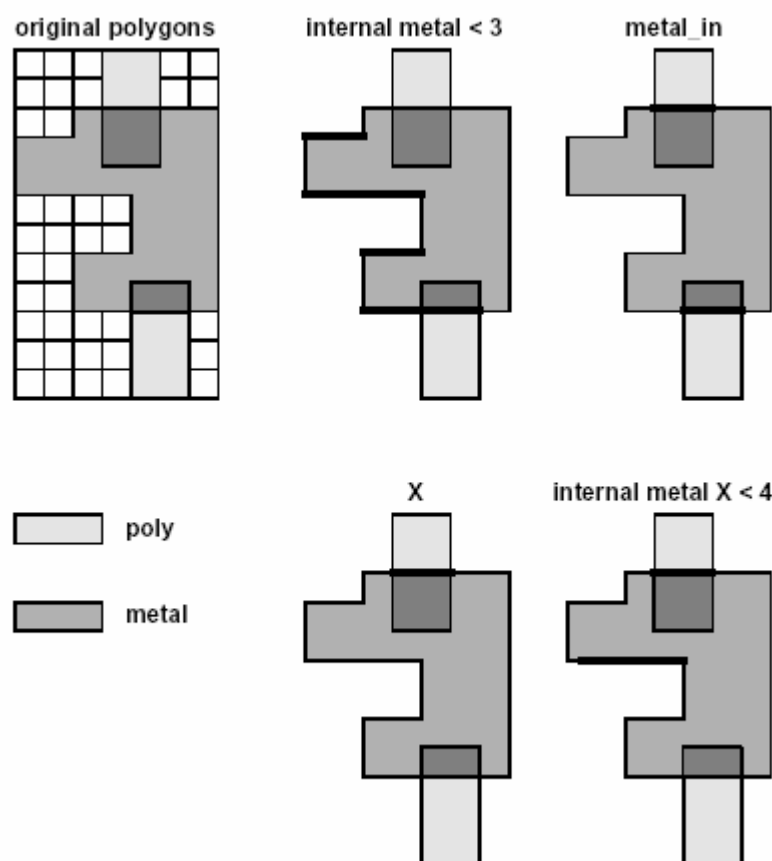
Internal 检查一般用于宽度检查，例如金属的最小宽度：

Internal            Metal1   <   0.4

这条指令可以查处所有的宽度小于 0.4 的金属 1。

T8.4 的右边是两个不同 layer 的 polygon 的 Internal 检查，请注意在 T8.4 中，只有箭头标出的边才可以检查。因为只有这一对是 interior edge。下面介绍了一个具体的例子来说明这条指令的应用。

```
// Metal Width in Poly Overlap Check
METAL004 {
@ The width of metal must be greater than or equal to 3 microns except
@ where poly overlaps metal by more than 1 micron, then the metal width
@ must be greater than or equal to 4 microns.
INTERNAL metal < 3 // normal rule
metal_in = INSIDE EDGE metal poly
X = INTERNAL (metal_in) poly <= 1
INTERNAL metal X < 4
}
```

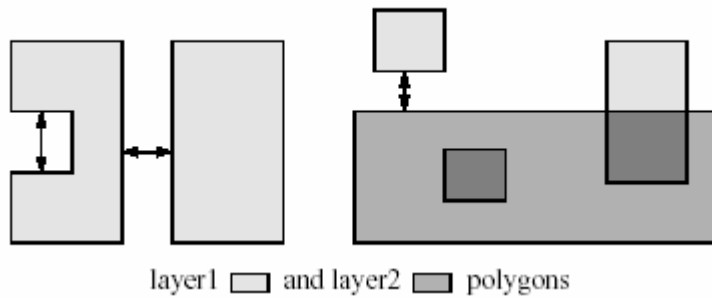


#### T8.5 Metal Width In Poly Overlap check

从上面的描述可以看出要检查的几何规则是一个比较复杂的规则，因此，这条规则用了四步来完成。

#### 8.3.2 External

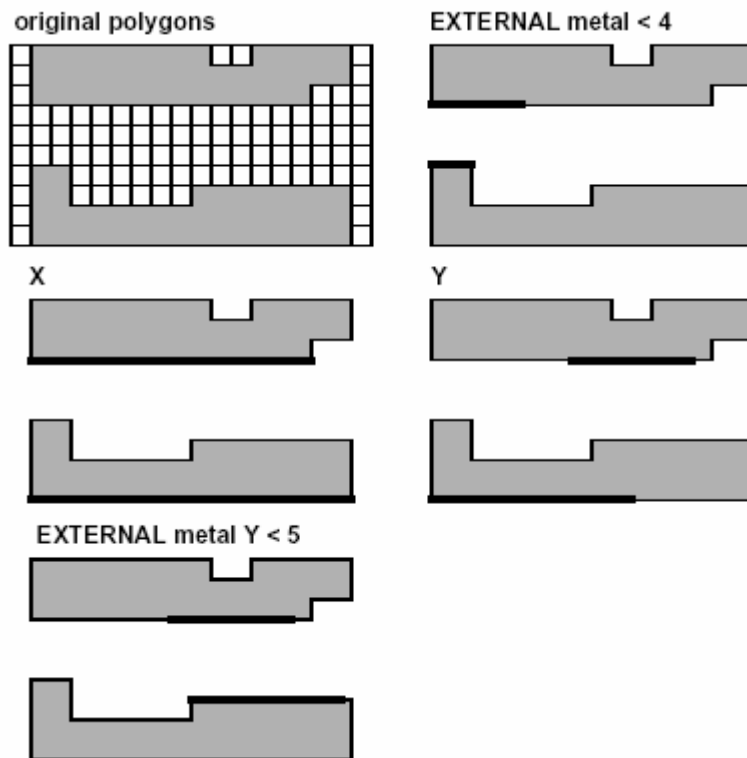
External 指令一般用于检查 polygon 外间距，可以用来检查同一 layer 的 polygon 的外间距、或检查两个不同 layer 的 polygon 之间的外间距。如图 T8.6 所示：



#### T8.6 External operation

T8.6 的左边是同一 layer 的 polygon 的外边 (exterior edge) 的检查, 右边是不同的 layer 的 polygon 的外边 (exterior edge) 的检查。理解 External 指令的关键是 External 指令是 polygon 的外边 (Exterior edge) 的相对关系, 只有 Exterior edge 之间才能检查。下面是一个 External 指令应用的例子。

```
metal_spacing {
  @ Metal to metal spacing must be >= 4 microns except where:
  @ (1) the metal edge length is greater than 10 microns, and
  @ (2) the width in case (1) < 3 microns.
  @ In the latter case, the metal spacing to edges satisfying both
  @ (1) and (2) must be >= 5 microns.
  EXTERNAL metal < 4 // Normal rule.
  X = LENGTH metal > 10
  Y = INTERNAL metal [X] < 3
  EXTERNAL metal Y < 5
  // Note that metal, X, and Y have the same layer of origin.
}
```

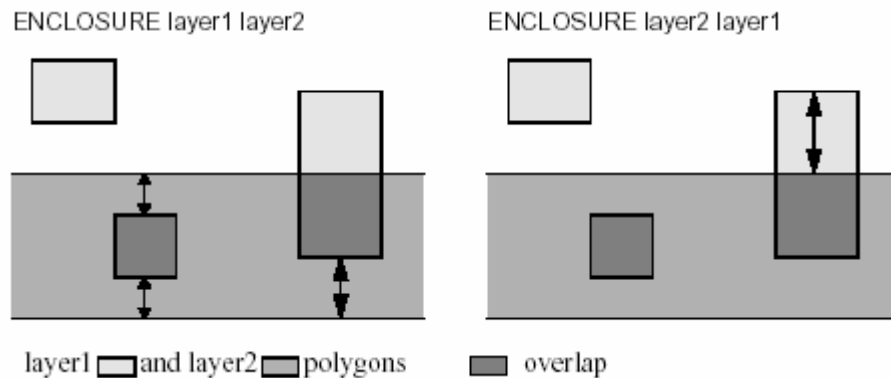


#### T8.7 Metal Spacing Check

T8.7 描述了金属的间距检查。请注意分析 External 的特性。

### 8.3.3 Enclosure

Enclosure 指令一般用于检查 polygon 的 overlap, 可以用来检查两个不同 layer 的 polygon 之间的关系。如图 T8.8 所示:

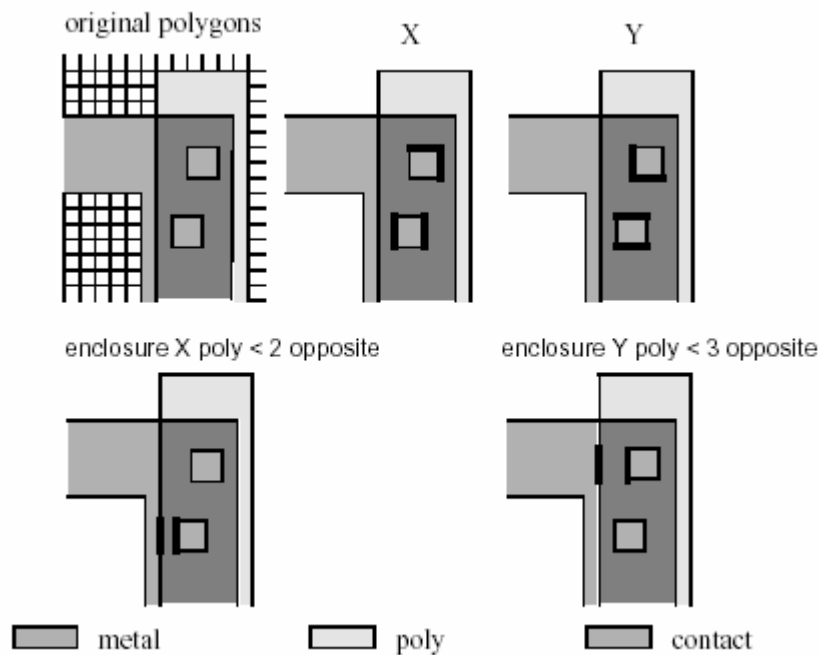


T8.8 Enclosure operation

Enclosure layer\_a layer\_b 是 layer\_a 的内边 (interior edge) 与 layer\_b 的外边 (exterior edge) 的关系, 下面是 enclosure 应用的例子。

```
DIR02 { // DIRECTION OF CURRENT FLOW CHECK
@ Enclosure of metal contact by poly must be greater than or equal to 3
microns in the
@ direction of current flow in metal; otherwise the enclosure must be greater
than or equal to
@ 2 microns.
@ Current flow direction in metal is defined as any metal direction where
metal surrounds
@ contact by more than 3 microns.
// X = edges not in current flow direction
// Y = edges in current flow direction
X = ENCLOSURE [ contact ] <= 3 metal OPPOSITE
Y = ENCLOSURE ( contact ) <= 3 metal OPPOSITE
ENCLOSURE X poly < 2 OPPOSITE
ENCLOSURE Y poly < 3 OPPOSITE
}
```

每一步的运算结果在 T8.9 中显示:



#### T8.9 Direction Of Current Flow Check

上面介绍了 Internal、External、Enclosure 三条指令，剩余的指令请参阅 calibre 的文档。calibre 的指令集十分复杂，足以完成各种规则的 DRC 检查，designer 要理解所有的指令才可以用好 calibre。

下一节介绍 calibre 做 DRC 的具体操作步骤。

#### 8.4 run calibre DRC

这一节介绍 calibre 具体的运行步骤。

##### Step1: 准备 Rule File 和 GDS File

Rule File 可以从流片厂家得到，或者 designer 根据流片厂家的设计规则来自己写。GDS 文件是由 APR 工具生成或 designer 手工画出的需要验证的版图。

##### Step2: 运行 calibre DRC

在 UNIX shell 下运行：

```
calibre -drc $path_to_rule_file/rule_file
```

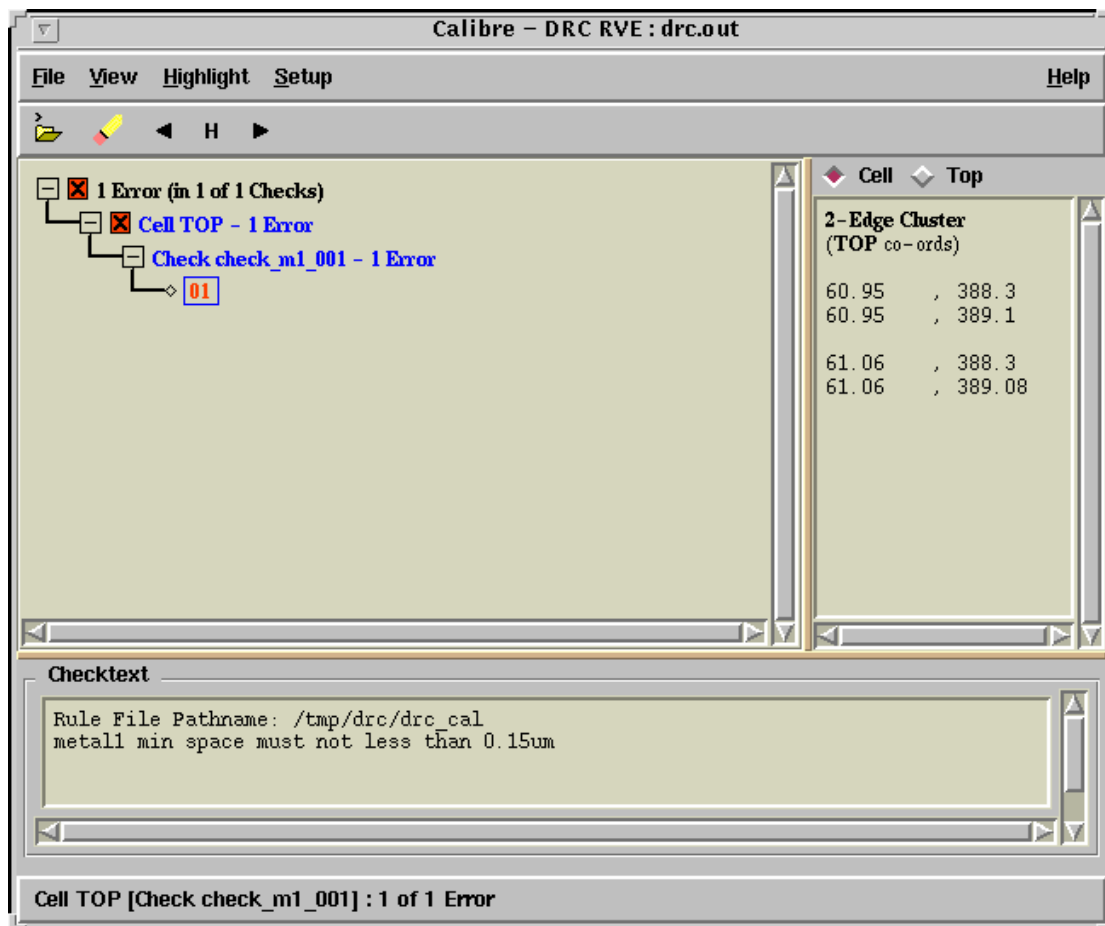
calibre 将会根据 rule\_file 找到需要验证的版图和验证规则进行 DRC 验证，结果存放在 rule\_file 指定的地方。

##### Step3: 分析 DRC 结果，修正错误。

接下来应该分析 DRC 的结果并改正查出的错误。现在需要用 cadence 的版图编辑工具 virtuoso。

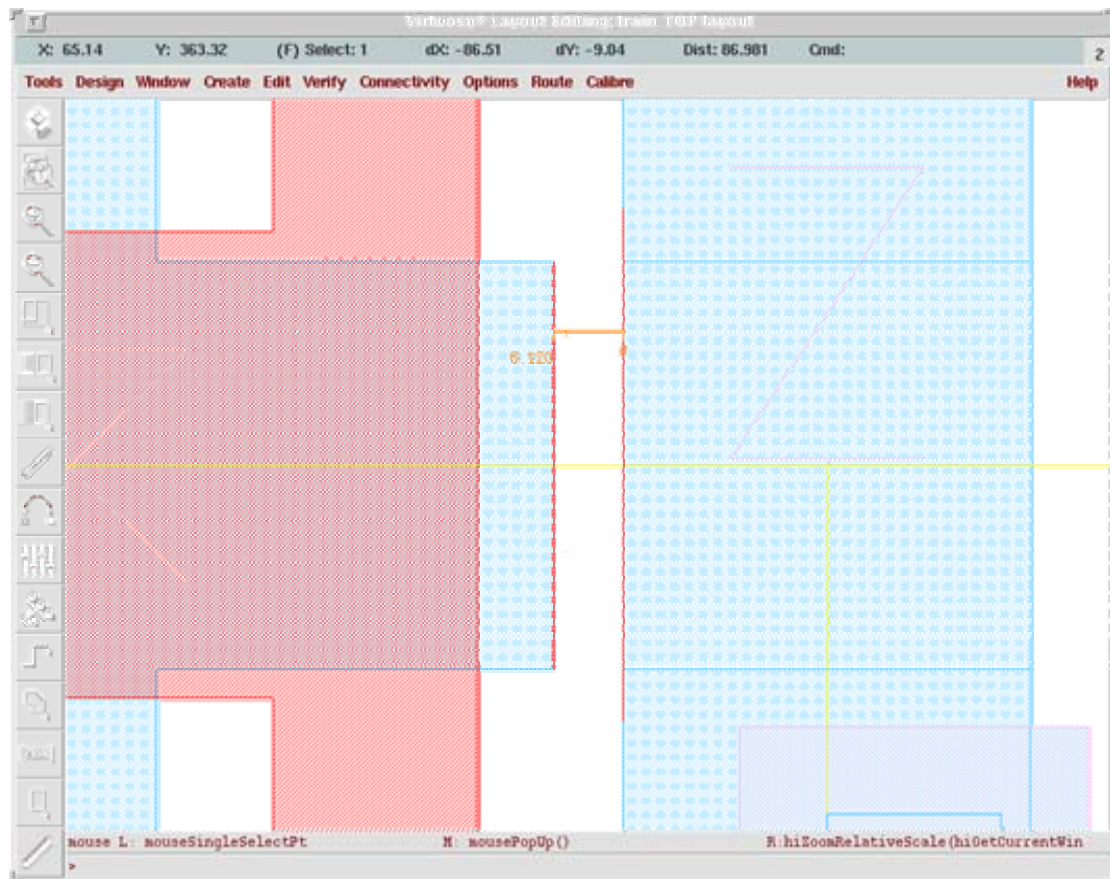
首先，运行 cadence 的 virtuoso，在 virtuoso 内调用 calibre 界面，如图 T8.10 所示，T8.10 显示版图有一个错误，错误提示是“metal1 min space must not less than 0.15um”，错误应该是 metal1 的最小间距不满足，在 calibre 内选择 highlight，virtuoso 将会高亮显示错误的区域，如图 T8.11 所示。



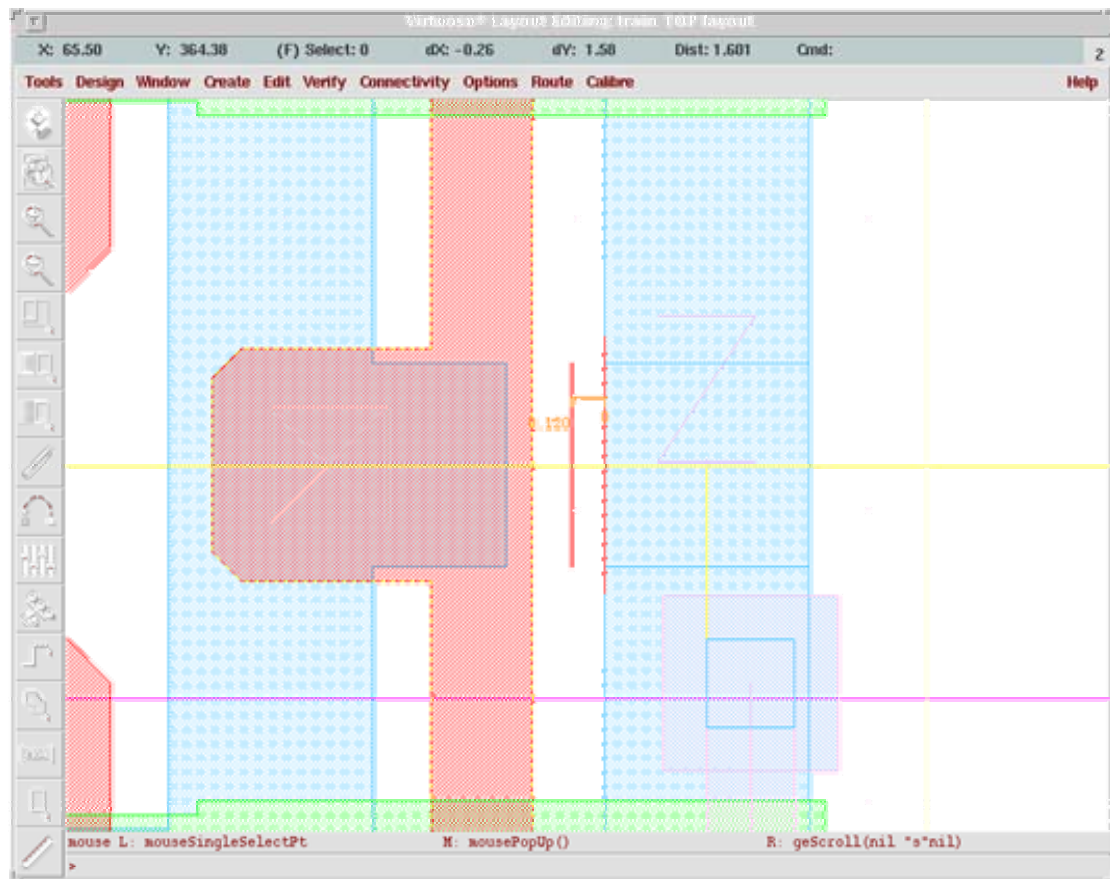


T8.10 calibre error report interface

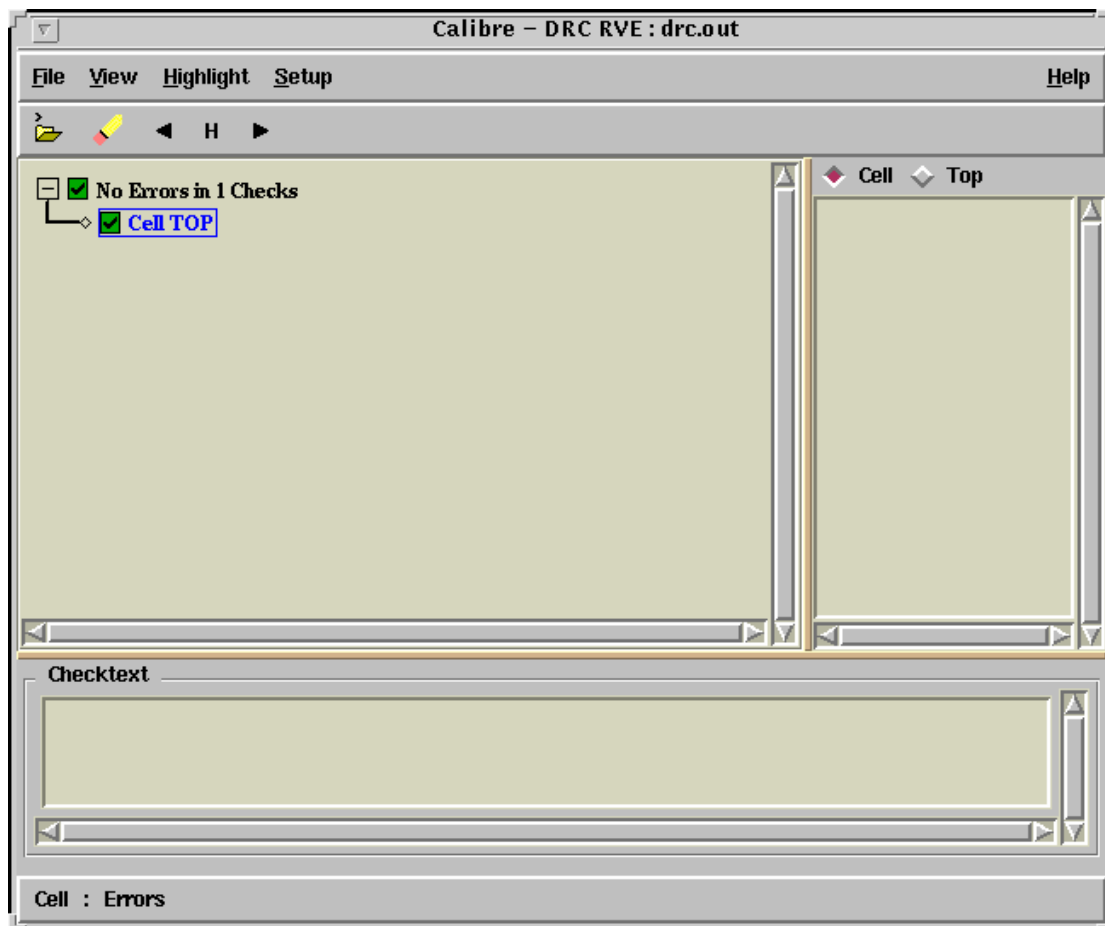
T8.11 中显示了版图出错的区域，分析可以得知 metal1 的间距小于 0.15um，修改版图如图 T8.12 所示，修改完版图，把改正的版图用 GDS 格式输出，用 calibre 再作一次 DRC 检查，检查结果如图 T8.13 所示，结果是版图通过了 DRC 检查，版图的 DRC 检查改正工作完成。



T8.11 Highlight error in virtuoso



T8.12 fix the DRC error in virtuoso



T 8.13 fix all error , DRC finish

改正所有的 DRC 错误后，就完成了 DRC 任务，这个通过 DRC 检查的版图在制造时就不会出现错误。

下一节介绍LVS（Layout Versus Schematic）

#### 8.4 LVS（Layout Versus Schematic）概述

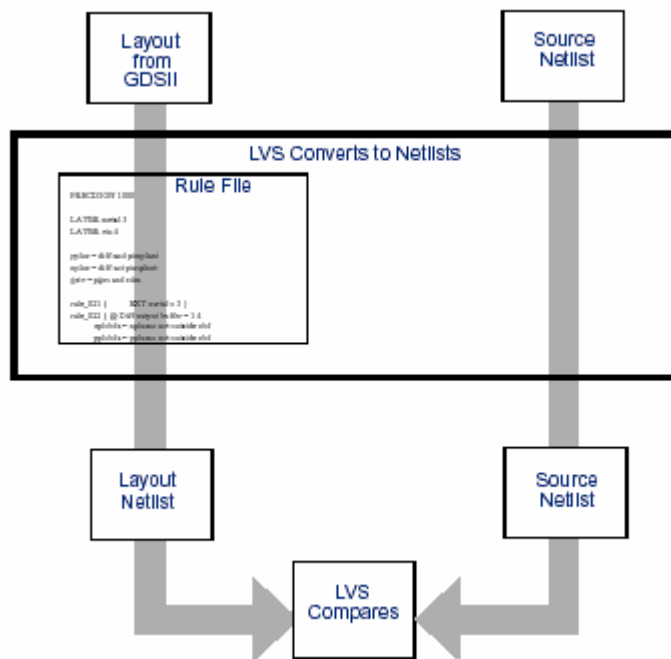
为什么要做 LVS? LVS 作了些什么?

APR 工具作出的版图，基本上是正确的。但是，不能保证 APR 工具是绝对正确的，也就是说，APR 工具可能会出错。为了发现并改正 APR 工具布线的错误，需要做 LVS。LVS 的目的就是证明版图的逻辑与网表是一致的，APR 将网表实现成了版图，现在 LVS 要证明这个实现是正确的。

LVS 的流程是：

- 1， 从 APR 工具实现的版图中提取（extract）出一个网表。
- 2， 将这个提取出的网表与 APR 工具要实现的网表比较，它们应该相同。
- 3， 如果两个网表不一样，这时手工修改版图，使之与要实现的网表相一致。

图 T8.14 描述了 calibre LVS 的 flow。



T8.14 LVS flow

Calibre 的 LVS 也是通过 Rule File 来控制 calibre 做 LVS 的，下一节介绍 calibre LVS 的 Rule File。

## 8.5 LVS Rule File

下面是一个简单的 calibre LVS Rule File，右边是对它的解释。

```

LAYOUT SYSTEM GDSII          # 指定 layout 的格式为 gds
LAYOUT PATH "/user/joeb/proj/layout/ictr" # 指定版图的路径
LAYOUT PRIMARY CalibreM      # 版图的 top cell
SOURCE SYSTEM SPICE          # 指定 source 的格式为 spice 网表
SOURCE PATH "/user/joeb/proj/logic/ictr" # source 的路径
SOURCE PRIMARY CalibreM      # source 的 top cell
MASK RESULTS DATABASE CalibreM.db # LVS 的结果保存文件，将来指导改正版图
LVS REPORT CalibreM.rpt      # report 文件
layer PWELL 1                # 读入版图的各层信息
layer OXIDE 2
layer RES 3
layer POLY 4
layer NPLUS 5
layer PPLUS 6
layer CONTACT 7
layer METAL1 8
layer VIA 9
layer METAL2 10
chip = extent
  
```

```

nsub = chip not pwell          # 进行 layer operation。
gate = oxide and poly
sd = oxide not poly
ngate = gate and nplus
pgate = gate and pplus
psd = sd and pplus
nsd = sd and nplus
sdm = pplus or nplus
connect psd pwell             # 定义连接关系
connect nsd nsub
connect metal1 poly nsd psd by contact
connect metal2 metal1 by via
device mn ngate poly nsd nsd pwell [0]    # 提取 nmos
device mp pgate poly psd psd nsub [0]     # 提取 pmos

```

calibre 根据这个 Rule File 提供的信息, 首先从版图中提取出一个网表, 再将这个网表与 APR 要实现的网表进行比较, 判断 APR 生成的版图是否正确。

这个 Rule File 首先设置 LVS 的运行环境, 然后读入版图各层的信息, 接下来进行 layer 的运算, 为下一步提取 device 作准备。

Connect layer1 layer2 是定义连接关系, 提取版图时将根据这个信息来提取连接关系。device mn ... 是提取具体的 mos 管,

```

device mn ngate ipoly nsd nsd pwell [0] //nmos transistors
      ↑   ↑   ↑   ↑   ↑   ↑   ↑
      seed gate source drain bulk
      Optional Trace Property Value

```

mn 指提出的 device 为 nmos 管, ngate 的解释为 seed, 即只要在版图上发现 ngate, 就提出一个 nmos 管, seed 后面的四个参数为 G S D B ,分别为 nmos 管的 4 各 pin。

根据 device 指令, calibre 从版图中提出各种 device, 根据 connect 指令提出连接关系。提出完整的 netlist 后, calibre 会与 source netlist 比较, 来判断网表与版图是否一致。

Calibre LVS 的步骤与 DRC 相似, 具体步骤参见 calibre 文档。