数模混合电路设计流程

马昭鑫 2012/5/23

本文主要面向模拟电路设计者，讲解了从行为级代码形式的数字电路到数模混合版图之间的流程，默认模拟版图和数字电路的行为级代码、testbench 已经完成。阅读者需确定自己会编写 Verilog 或 Spice 格式的网表，熟悉 Linux 的文件操作，了解 Spectre、Virtuoso、Calibre、Modelsim、Design Compiler（dc）、Astro 等 EDA 工具的使用方法。

由于本人才疏学浅，经验不足，难免会在文中出现一些错误，恳请高手给予指正。

数模混合电路的仿真方法

一般的设计流程中数字电路和模拟电路是分开进行设计的，但有些时候希望能将数字电路和模拟电路放在一起仿真来验证设计，这就需要用到混合电路的仿真方法。在 Cadence 工具中有专门用作混合电路仿真的仿真器 spectreVerilog，其实现方法是首先将模拟模块与数字模块区分开并设置接口电平，然后在 ADE 中设置数字电路的测试代码，调用不同的仿真器分别对数字模块和模拟模块进行仿真，最后将结果汇总显示或输出。

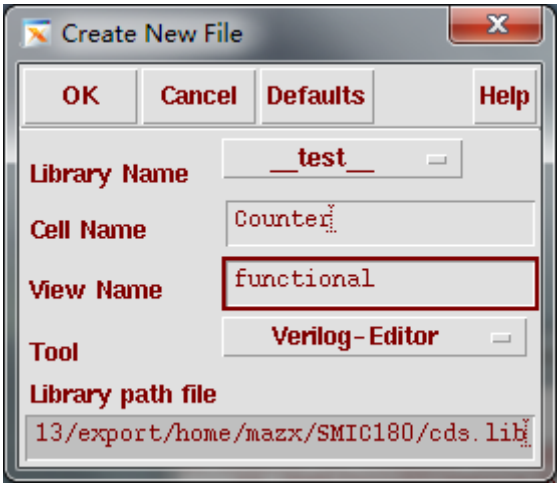
下面将以一个简单实例的形式讲解混合电路的仿真方法。

一、 建立数字模块

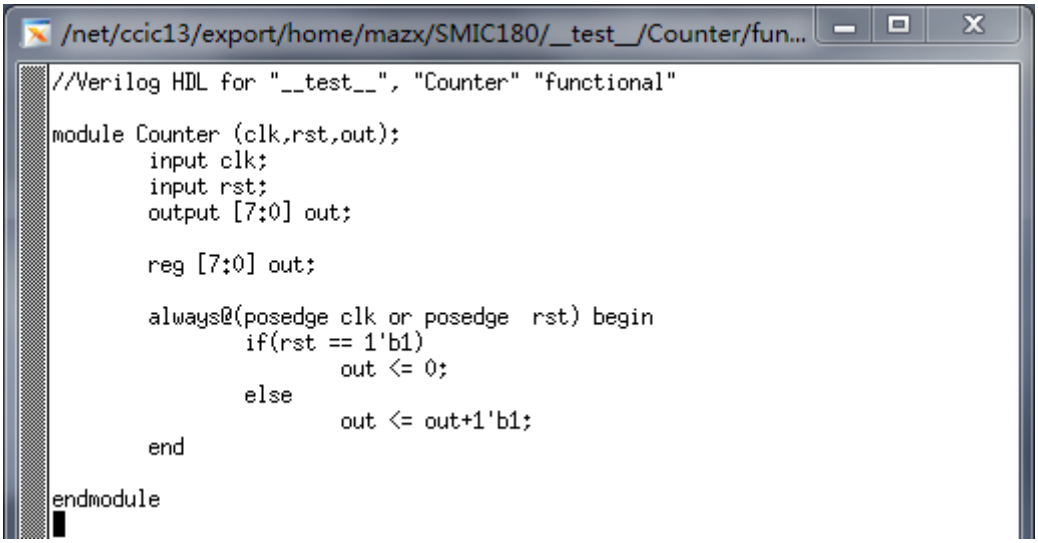
①在命令行中输入下面的命令设置 NC-Verilog 和 Cadence 并启动 Cadence；

```
setdt ldv
setdt ic
icfb&
```

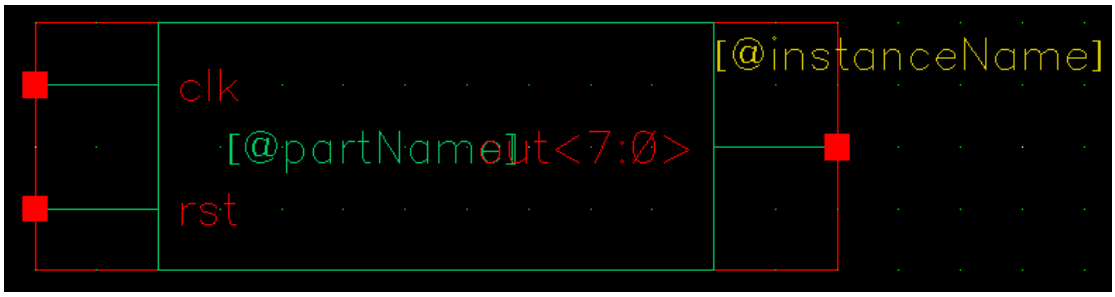
②建立 Library 的方法不再累述，创建 Cell view 时注意 Tool 选择 Verilog-Editor，View Name 填写 functional；



③点击 OK 后会弹出有模块代码框架的 vi 窗口，将设计需要的代码输入或粘贴进去；



④保存并关闭后如果没有错误会弹出创建 Symbol View 的询问对话框，确定后会进入 Symbol 编辑器，并自动生成了 Symbol（注意在 Cadence 中总线用尖括号<>表示）；

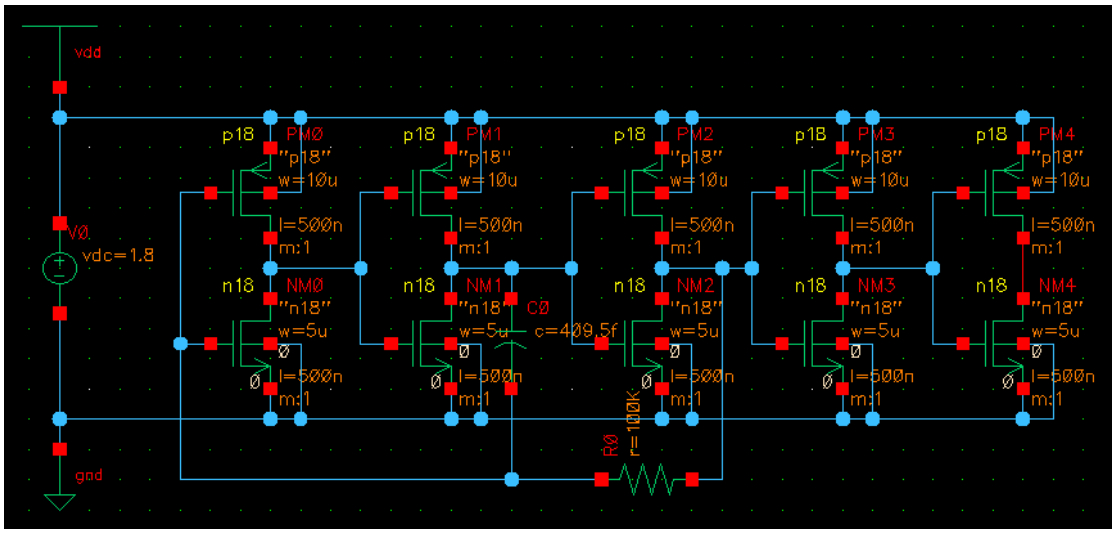


⑤保存并关闭 Symbol 编辑器。

至此已经完成了数字模块的创建。

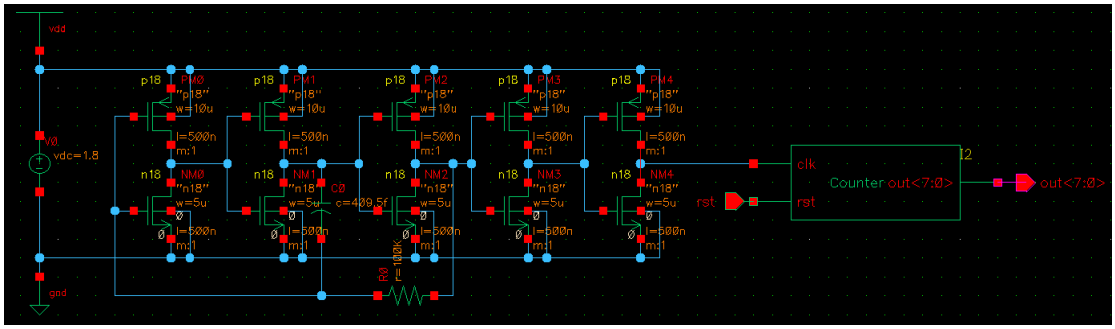
二、 建立模拟模块

模拟电路的创建方法无需赘述，这里搭建了一个输出频率为 10MHz 的环形振荡器。

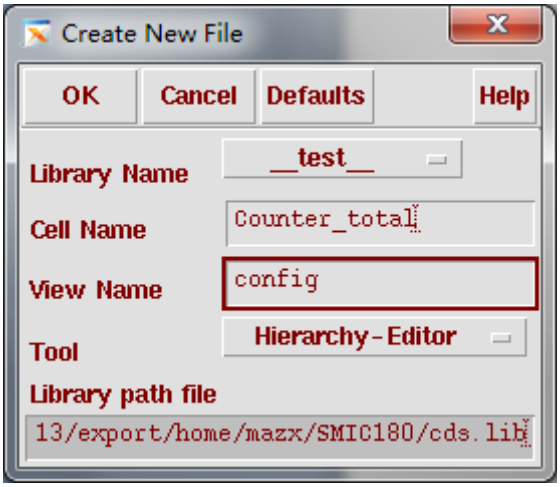


三、 配置混合电路

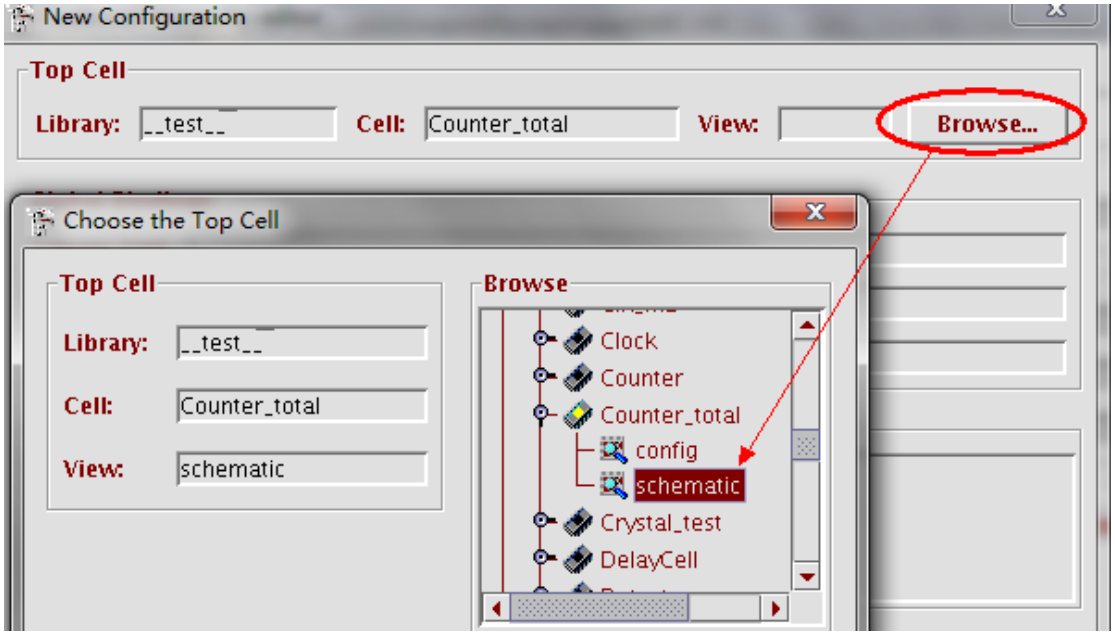
①直接在模拟电路中搭建混合电路，将数字模块添加进来，连接数模之间的接口，预留给 testbench 的接口用 port 连接，Check&Save 后关闭 Schematic 编辑器；



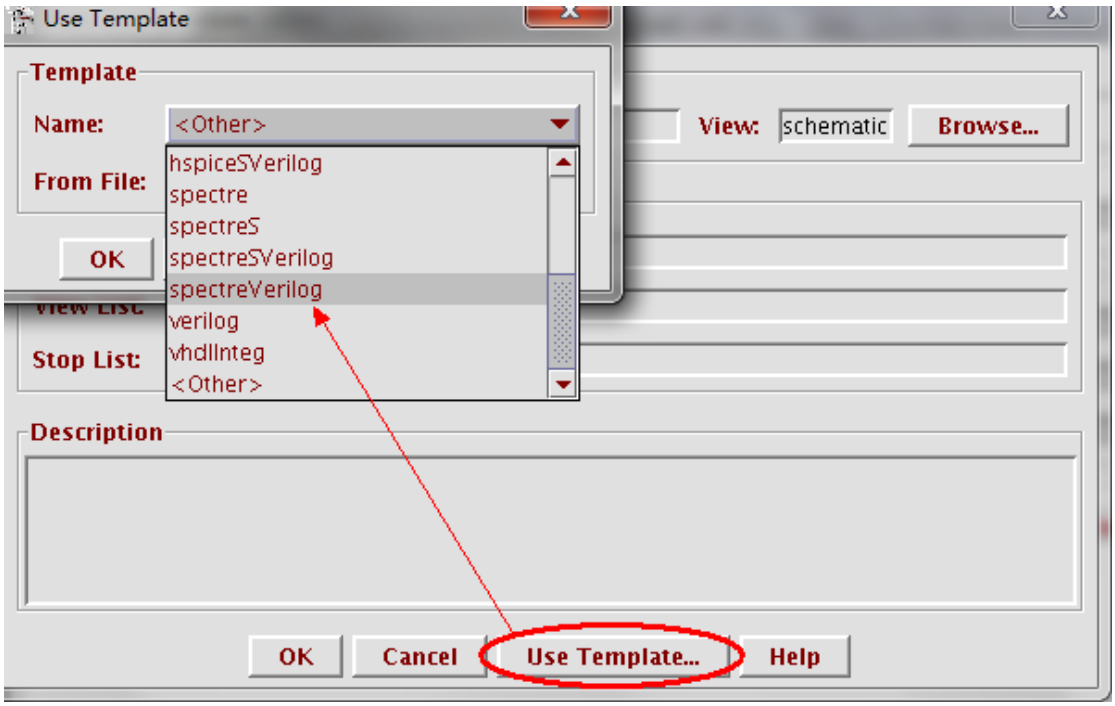
②新建与混合电路模块同名的 Cellview，Tool 选择 Hierarchy-Editor，View Name 会自动修改成 config；



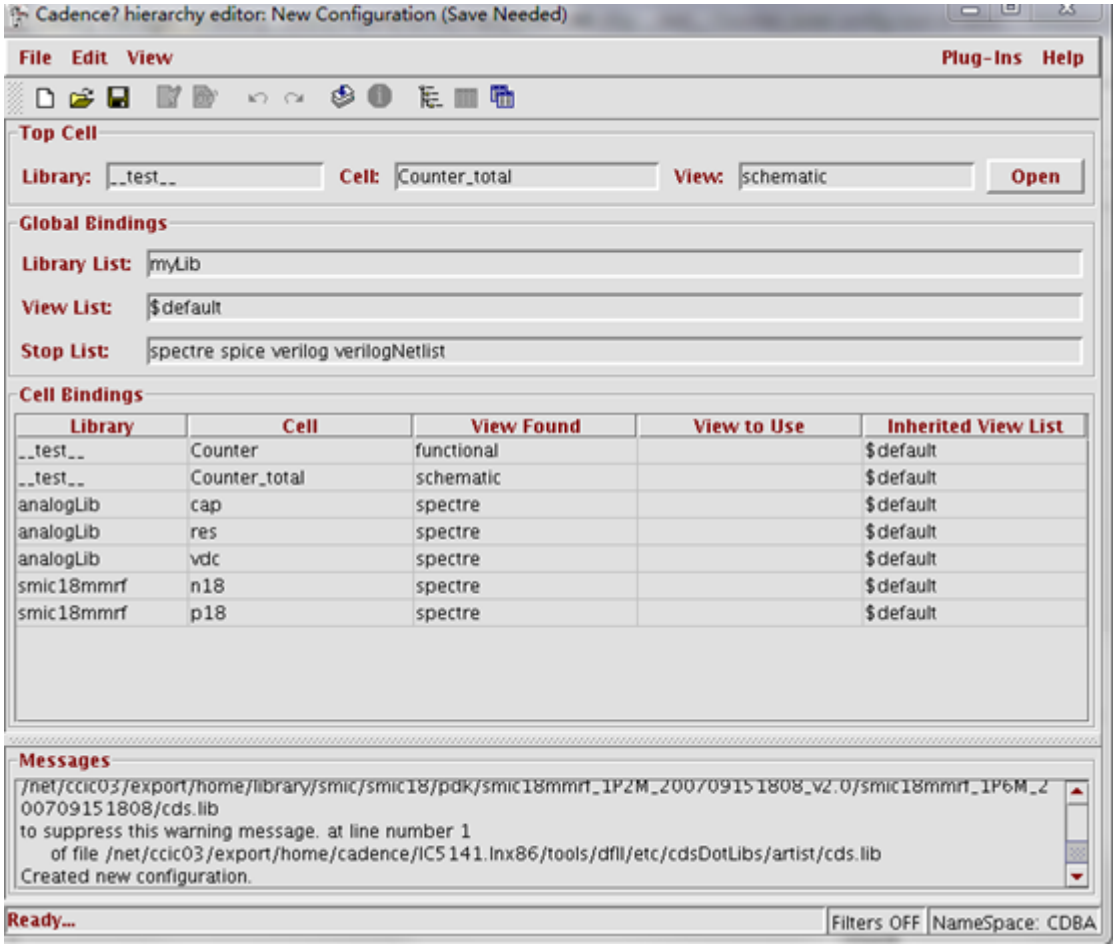
③点击 OK 后弹出 Hierarchy 编辑器的界面，点击 Browse，选择混合电路的 schematic cellview，然后点击 OK；



④点击下方的 Use Template 按钮，在弹出的对话框中将 Name 选择为 spectreVerilog，点击 OK 回到原对话框，再点击 OK 进入主界面；



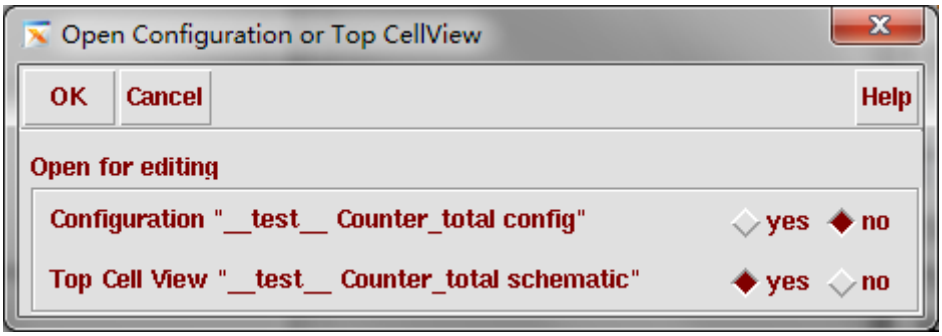
⑤在主界面可以看到不同的 Cell 自动选择了不同的 View，保存退出。



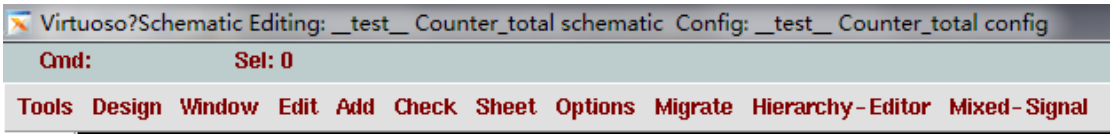
这时可以在 Library Manager 中看到 config 的 view，它包含了混合电路的配置信息。

四、 混合电路仿真

①双击 config view 或右击 open，弹出打开方式的对话框，config 选择 no，schematic 选择 yes，点击 OK；

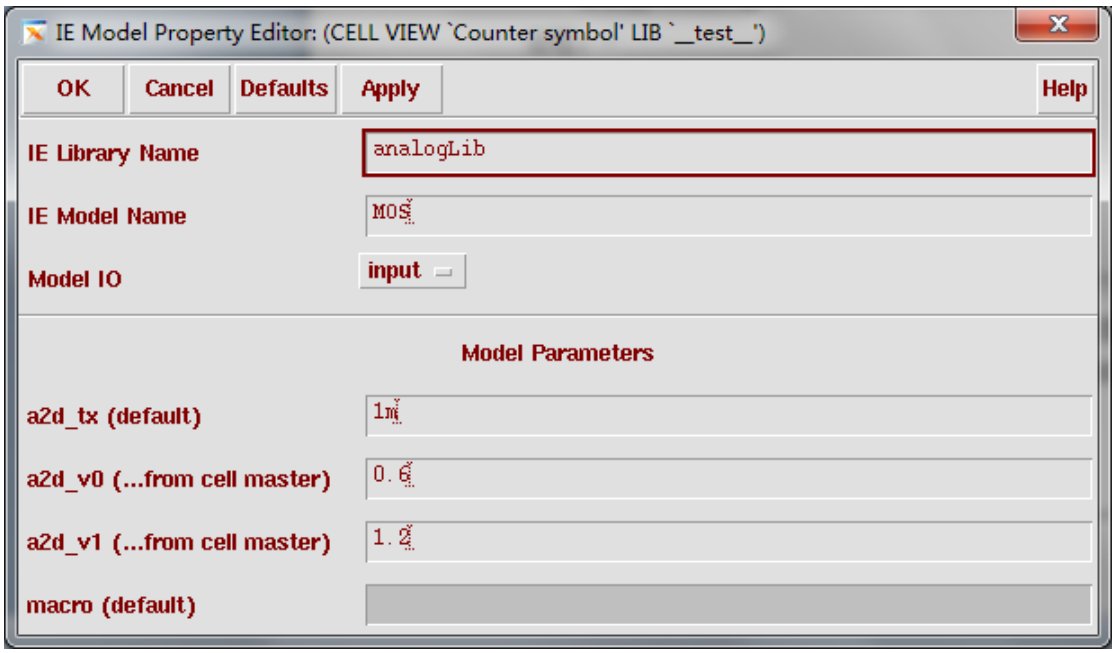


②在打开的 Schematic 编辑器中点击 Tools→Mixed Signal Opts.，会发现编辑器的菜单栏变成了如下的样子：

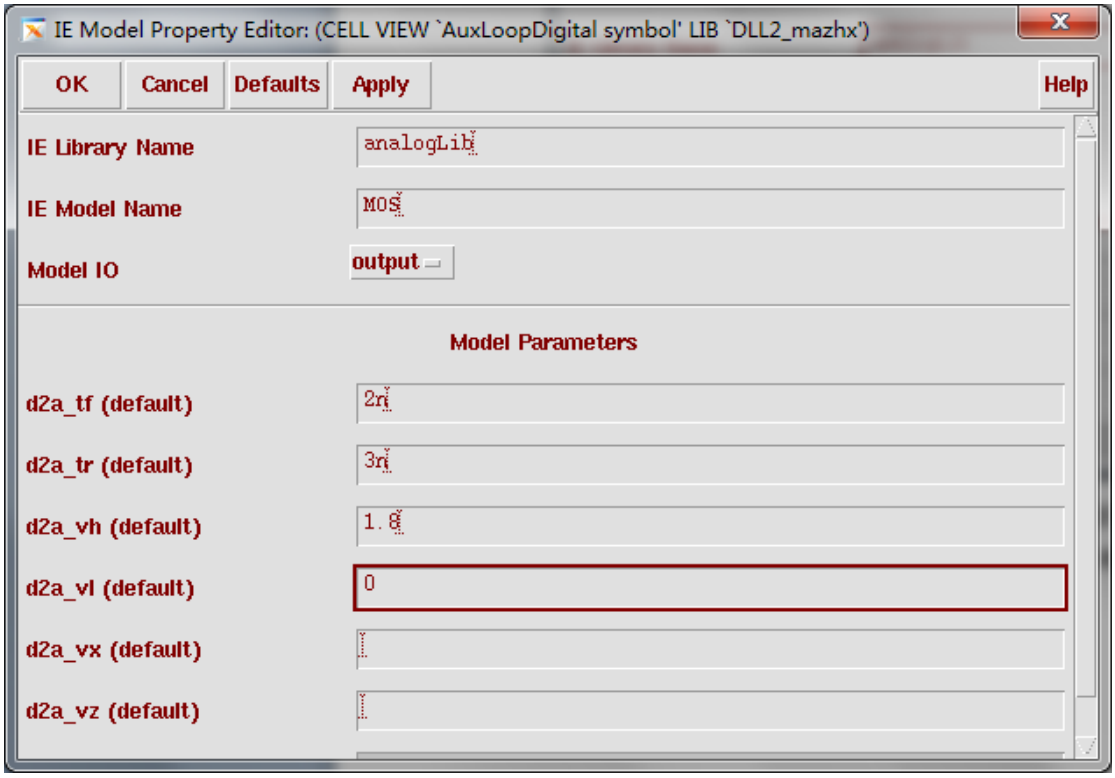


③点击 Mixed-Signal→Display Partition→All Active，如果弹出对话框则点击 OK，可以看到数字模块和模拟模块用不同的颜色标注了出来；**（这步不是必需的）**

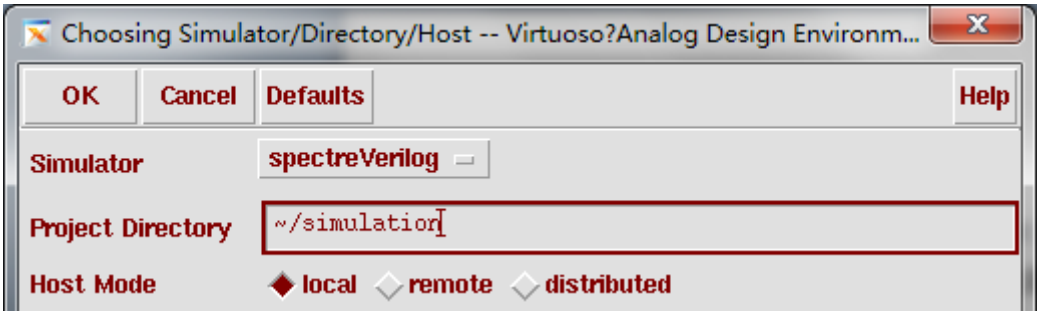
④点击 Mixed-Signal→Interface Elements→Cell，点击数字模块，弹出设置接口电平的对话框，需要设置 a2d_v0 和 a2d_v1，这两个参数规定了模拟转成数字时的电平，对于 1.8V 电源电压可以分别设置为 0.6 和 1.2；



⑤如果有数字输出给模拟的接口，需要将该对话框中的 Model IO 改成 Output，设置 d2a_vh 和 d2a_vl 两项，即数字高低电平对应的模拟电压；



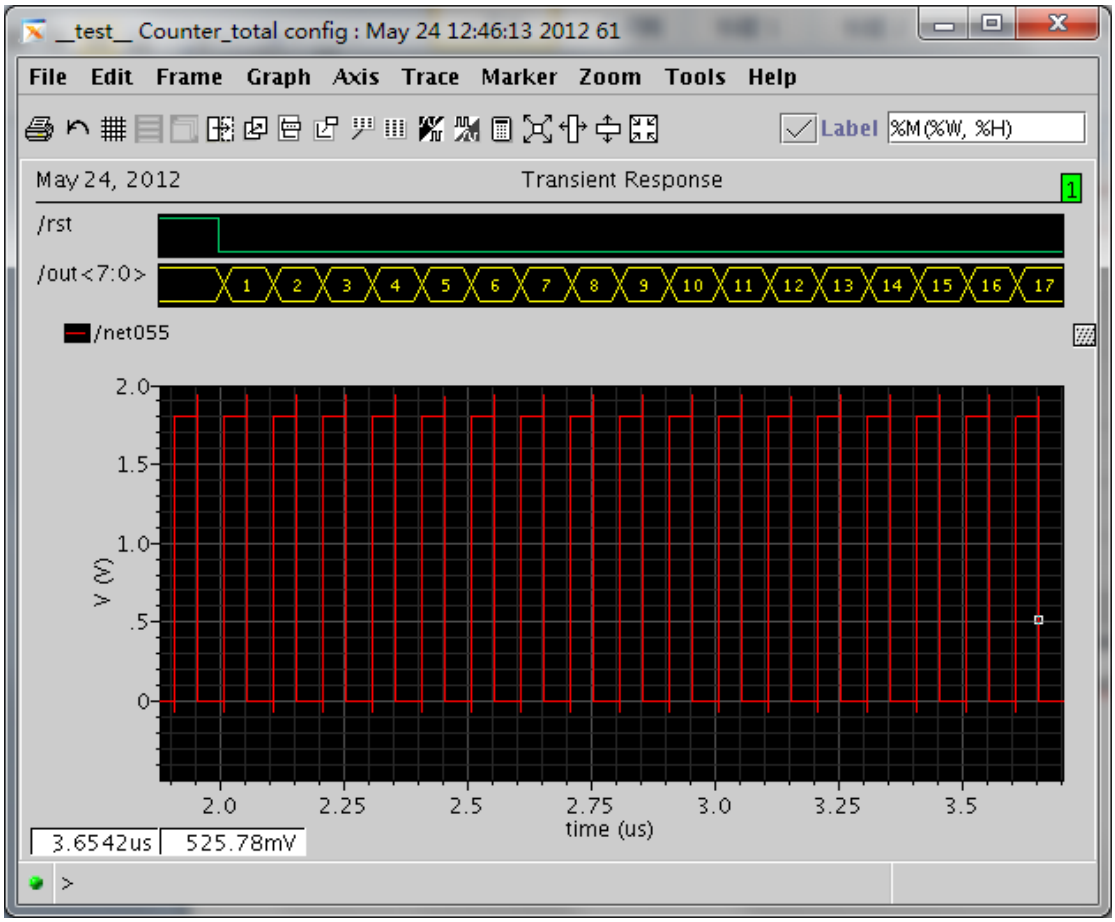
⑥进入 ADE，点击 Setup→Simulator/Directory/Host，在弹出的对话框中将 Simulator 选择为 spectreVerilog，点击 OK 确定退出；



⑦点击 Setup→Stimuli→Digital，弹出设置 testbench 的 vi 窗口，将设计需要的 testbench 输入或粘贴进去，保存并退出；

⑧设置瞬态分析并选择需要 plot 的信号；

⑨点击 Netlist and Run 后等待仿真结束就会弹出信号的显示窗口，双击总线名可以以不同的进制显示数据；



数模混合电路的仿真至此结束。

参考文献

[1] http://wenku.baidu.com/view/580419620b1c59eef8c7b42e.html, Cadence_实验系列 12_数模混合电路设计_spetreVerilog, 中山大学

[2] http://wenku.baidu.com/view/821e7de881c758f5f61f67c8.html, 用 SpectreVerilog 进行模数混仿

数字电路设计流程

用 Verilog 写出实现了功能的数字电路仅仅是一个开始，到真正可以流片的版图还有很多流程要走，综合可以把行为级电路转换成标准单元组成的电路，布局布线可以把这些标准单元按照一定的方式放置，并用金属线一一连接。但并不是每一步产生的结果都符合我们的要求，一再的仿真验证是必不可少的。

一、 行为级仿真

工具：Modelsim
输入文件：行为级模块（*.v）、testbench
输出文件：无
目的：检验设计是否能实现所需功能

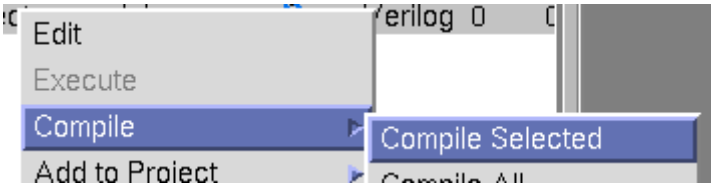
① Modelsim 的启动方法是在命令行中输入下面的命令：

```
setdt ams
vsim&
```

② 创建 Project，并将行为级模块（*.v）和 testbench 添加到 Project 中

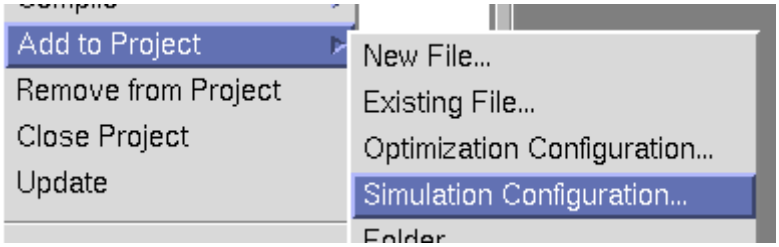
③ 编译（Compile）

选中要编译的文件，右键点击后选择 Compile→Compile Selected，如果代码没有问题可以看到下方代码框中显示 successful，Workspace 中文件的 Status 会变成绿色的对号；否则会提示 error，Status 变成红色的错号，这时双击错误提示信息可以在弹出的窗口中看到详细的编译失败的提示。



④ 配置仿真（Simulation Configuration）

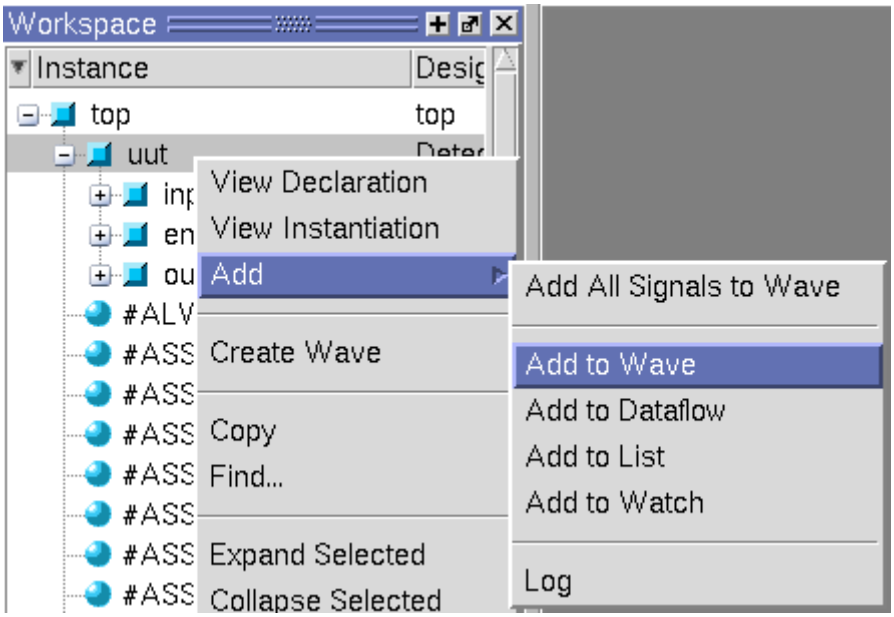
仿真的方法有两种：一种是在 Library 中找到 testbench 的顶层模块名，右击选择 Simulate；另一种是在 Workspace 文件列表的空白处右击，选择 Add to Project→Simulation Configuration，在弹出的框中选择 testbench 的顶层模块名，然后在 Libraries 中添加编译产生的 Library，最后 Save。后者会产生一个配置文件，使用起来较为方便，建议将 Simulation Configuration Name 改成有意义的文字，比如 Simulation Behav，以便于与其他的仿真配置文件区分。双击这个文件就可以进入仿真界面。



⑤ 添加信号并仿真

在 Workspace 中 Instance 列表中选择要查看的元件，右击选择 Add→Add to Wave 可以看到元件内部的信号线都加到了 wave 窗口中，设置合适的时间点击工具栏上按钮 

就可以进行仿真。



⑥ 观察仿真结果

在 wave 窗口中可以放大、缩小、选择合适的进制显示波形，根据仿真结果观察是否能完成所需功能。

二、 综合

工具：Design Compiler（dc）

输入文件：行为级模块（*.v）、工艺文件（slow.db，SP018W_V1p7_max.db，dw01.sldb，dw03.sldb、综合宏脚本（*.tcl）

输出文件：网表文件（*.sv）、标准延时文件（*.sdf）、标准延时约束文件（*.sdc）

目的：给模块中的线路添加约束，将行为级模块转换成网表级模块，并产生延时文件等

综合的脚本文件（*.tcl）是本过程的核心，该文件是一系列 dc 命令的集合，主要包括：设置工艺库（*.db 等），读入行为级模块文件（*.v），分析检查等，设置输入驱动，设置输出负载，创建时钟并设置时钟的物理属性如周期、延迟、不稳定程度等，设置时钟域，设置输入输出延时，设置扇出数，综合，生成一系列文件（*.sv、*.sdf、*.sdc 等）等。综合后需要生成 timing report 检查是否有违例（violation），如果有违例出现需要修改综合脚本后重复该过程直到所有的路径都通过。如果数字模块规模比较大，脚本编写的好坏与否会严重影响电路的性能甚至功能，建议在数字后端高手的指导下编写。

dc 的启动方法是在命令行中输入下面的命令（其中 run.tcl 为综合脚本的文件名）

```
setdt syn
dc_shell-t -f run.tcl
```

* 关于违例我的理解如下：从数据准备好到取数据的时钟沿到来之间的时间差叫做 slack，显然 slack 应该为正以确保取到的数据是有效的,如果路径延时过大造成 slack 为负时违例就产生了。这时可以查看产生违例的路径，通过优化设计、修改约束的方法消除违例。值得注意的是 dc 对同步电路的综合能力很强，但对于异步电路违例的检查可能会出现问题（也可能是综合脚本的原因），对于异步电路中出现的某些违例可以经过慎重的检查忽略掉。

** dc 可能会对电路做一些很奇怪的优化，比较简单有效的解决办法是将原模块尽可能的分层次写成若干个小模块，将不同功能的电路在 module 的层次上分隔开。

三、 网表级仿真

工具：Modelsim

输入文件：网表文件（*.sv）、标准延时文件（*.sdf）、标准单元模型文件（smic18.v 或 smic18_neg.v）、testbench

输出文件：无

目的：检验设计在相关约束下是否能实现所需功能

网表级仿真与行为级仿真的区别在于用行为级语句描述的电路变成了由标准单元组成的电路，并且增加了延时这种具有物理属性的参数。

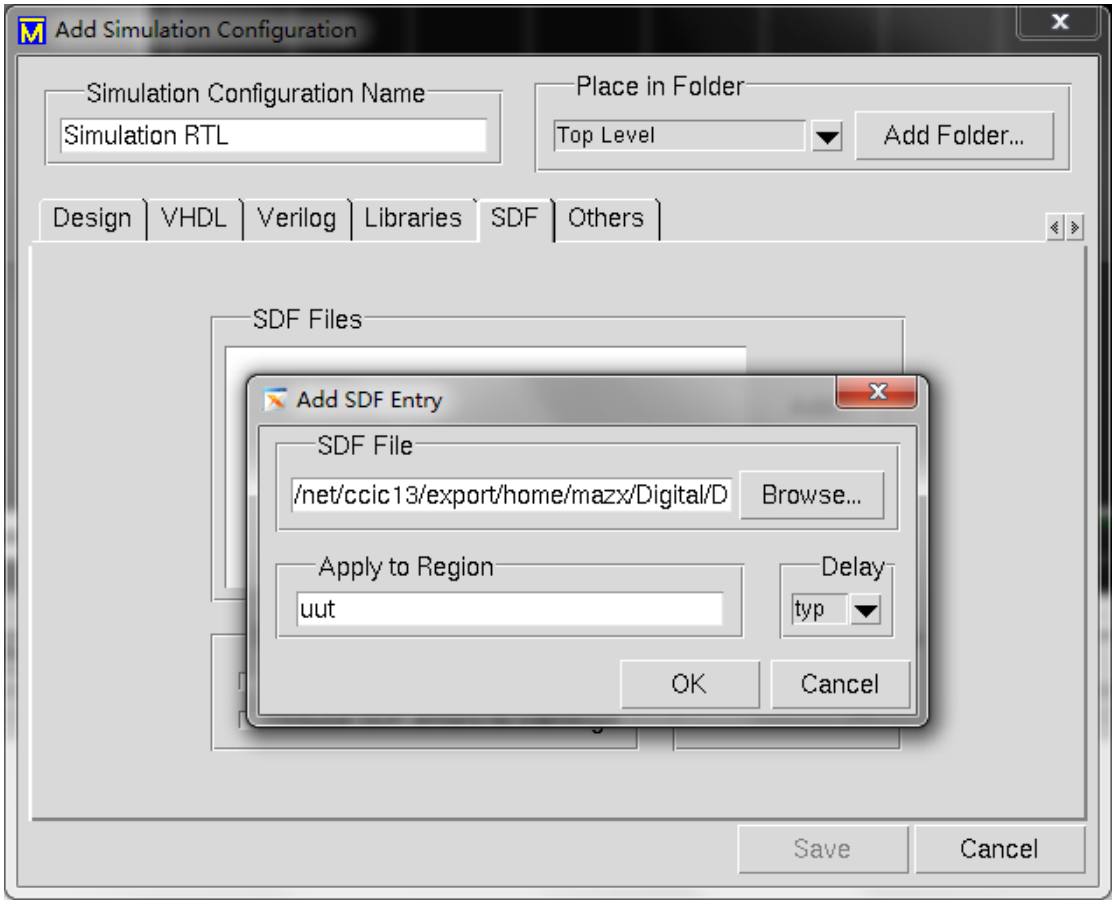
具体操作与行为级仿真相似，但注意由于网表文件（*.sv）的顶层模块名与行为级模块文件（*.v）中的顶层模块名是一样的，所以编译前先将行为级模块文件移除 Project；另外由于网表文件中引用了标准单元，所以要把标准单元模型文件（smic18.v 或 smic18_neg.v）也添加到 Project 中并编译。

标准延时文件（*.sdf）文件的添加是在 Simulation Configuration 中进行的，选择 Add Simulation Configuration 对话框中的 SDF 选项页，点击 Add，在弹出的对话框中选择 SDF 文件路径，需要注意 Apply to Region 必须要填写 testbench 中模块的例化名，比如 testbench 中例化的语句如下

```
MyModule uut (.clk(clk), .in(in), .out(out));
```

则该处要填写 uut。

仿真过程中可能会报一些 Error 或 Warning，一般是由于时钟沿与数据变化沿正好对齐导致某些数据不能确定导致的，在异步电路中更为常见，可以根据实际情况分析忽略某些 Error。



四、 布局布线

工具：Astro

输入文件：带有 pad 描述的顶层文件（*.v）、pad 布局文件（*.tdf）、网表文件（*.sv）、经过修改的标准延时约束文件（*.sdc）、标准单元库与 Pad 库文件（在 apollo/stdcell 与 apollo/iocell 目录下）、布局布线宏脚本（*.cmd）、Verilog 文件列表文件（vloglist）

输出文件：网表文件（*.vg）、标准延时文件（*.sdf）、GDS 文件（*.gds）

目的：将网表转换成版图并添加 pad，主要工作是布局与布线

在这一步首先要做的是规划 pad 布局，然后编写带有 pad 描述的顶层文件（*.v）和 pad 布局文件（*.tdf），这将关系到芯片最终的 I/O 接口布局，以及与模拟部分版图的拼接；然后编写布局布线脚本（*.cmd）在 Astro 中完成布局布线以及分析优化等工作。由于这部分工作比较多，下面分成三部分详细讲述。

1. 带有 pad 描述的顶层文件（*.v）

顾名思义这个文件包含了 pad 信息，除此之外还有例化的数字模块和预留给模拟电路部分的接口，如果需要在模拟部分与数字部分之间、电路部分与 pad 部分之间添加 buffer，也应该在这个文件中做描述。

pad 的类型以及用途见下表：

pad 类型	Cell Name	I/O	用途
模拟 pad	PANA1APW	PAD	模拟接口 pad
	PVDD1CAPW	SVDD1CAP	模拟电源 pad
	PVSS1APW	SVSS1AP	模拟地 pad
	PVDD5APW		ESD 电源 pad
	PVSS5APW		ESD 地 pad
数字 pad	PISW	C（内）、PAD（外）	数字触发输入 pad
	PIW	C（内）、PAD（外）	数字输入 pad
	PO16W	I（内）、PAD（外）	数字输出 pad
	PVDD1W	VDD	数字内核部分电源 pad（1.8V）
	PVDD2W		pad 上电平转换部分的电源 pad（3.3V）
	PVSS3W	VSS	数字地 pad
其他	PDIODE8W	VSS1、VSS2	二极管 pad，用作断开 pad 环做地隔离

其中需要注意的有：

①模拟 pad 上有若干道金属片，分别为 SAVDD 与 SAVSS 间隔连接，最终会通过 Filler 形成 pad 的 ESD 环接到 ESD 电源与 ESD 地上去；

②PISW、PIW 和 PO16W 带有电平转换功能，即 pad 外侧是 3.3V 电平，pad 内侧是 1.8V 电平，转换电路通过 PVDD2W 供电；

③数字 pad 环要与模拟 pad 环要通过二极管 pad 断开，不仅仅因为模拟地与数字地需要隔离，还因为数字 pad 和模拟 pad 上的金属环的连接方式是不同的，除了 SAVDD、SAVSS 还有 VDD、VSS、VDD33、FF。

在.v 文件中描述 pad 的时候模拟 pad 的 I/O 全部空白，数字 pad 要按连接方式填写，电源 pad 填`.VDD(1'b1)`，地 pad 填`.VSS(1'b0)`。这样 astro 会根据描述自动用金属线把电路与 pad 连好，电源和地也会根据相应的方式连接，而模拟部分需要后面在 Virtuoso 中手动连线。

最终.v 文件的形式应当如下：

```
module 模块名_pad( 预留给模拟电路的接口, 经过 pad 的数字测试接口 );
    // I/O 口声明;
    input ...;
    output ...;
    // 内部连线声明;
    wire ...;
    // 数字模块例化声明;
    模块名 例化名 ( ... );
    // Buffer 例化声明;
    CLKBUFX20 buf_...(.A(...), .Y(...));
    // 数字 pad 例化声明
    PIW     Pad_...(.C(...), .Y(...));
```

```
    PO16W   Pad_...(.I(...), .Y(...));
    PVDD1W  Pad_VDD_D(.VDD(1'b1));
    PVDD2W  Pad_VDD33();
    PVSS3W  Pad_VSS_D(.VSS(1'b0));
    // 模拟 pad 例化声明
    PANA1APW    Pad_...();
    PVDD1CAPW   Pad_VDD();
    PVSS1APW    Pad_VSS();
    // 隔离 pad 例化声明
    PDIODE8W    PAD_DIODE_1();
    PDIODE8W    PAD_DIODE_2();
endmodule
```

2. pad 布局文件（*.tdf）

.v 文件中只是对 pad 进行了例化声明，而 pad 位置的说明要在 pad 布局文件（*.tdf）中进行。该文件的格式比较固定，首先声明并放置 Corner Cell 作为 pad 的参考，然后设置一些变量控制 pad 的位置，最后放置 pad。

Corner Cell 的宽度是 210，pad 的宽度是 76，一般 pad 的间距选为 24，据此可以很容易通过套用模板写出适合自己的设计的 pad 布局文件。

唯一需要注意的是 pad 的顺序是 Left、Right 从下向上递增，Top、Bottom 从左向右递增。

3. 布局布线宏脚本（*.cmd）

该文件的作用与综合脚本（*.tcl）在综合中的作用是一样的，本质上都是一系列命令的集合。建议直接修改可用的脚本实现自己的在布局布线功能。

一般来说需要修改的地方有

①各种库的路径；

②libName、cellName，改成和自己的模块名一样；

③修改 Verilog 文件列表文件（vloglist），其中包含带 pad 描述的顶层文件（*.v）和 dc 生成的网表文件（*.sv）；

④Setup Floorplan：Core Width、Core Height、Core To Left、Core To Right、Core To Top、Core To Bottom 等参数根据数字模块在整个芯片中的位置与大小修改；

⑤Create Pins：在合适的位置创建预留给模拟电路接口的 Pin，最好根据模拟版图确定，以方便后续工作；

⑥Add Pad Fillers：需要根据芯片大小修改覆盖的面积，但貌似过大没有坏处，所以直接修改成（0,0）到（9999,9999）也可以；

⑦Create Ground&Power Rings：根据数字模块的大小和位置确定电源环的位置，这部分空间需要提前留出来，建议把电源环放在外面，以方便用 M1_NW 将数字模块与模拟模块的衬底隔离开来，如果数字电路规模比较大还需要通过同样的方法创建电源条；

⑧Place Standard Cells：根据电路逻辑特点选择 timing/routability 的值，数字越小表明

组合逻辑占的比例越大；

⑨Add Route Guides：根据模拟版图的位置、形状以及金属层禁止某些地方布线；

⑩如果使用了 Create Pins，在 GDS Output 的时候要把 Output Pins 的 As Text 和 As Geometry 选中，否则导出的 GDS 文件中没有 Pin，不方便连线。

一般情况下需要修改的地方有以上几种，实际操作中应当根据自己的设计和模板进行相应的修改。

除了布局布线脚本（*.cmd）以外还要写标准延时约束文件（*.sdc），这个文件是在 dc 生成的*.sdc 的基础上修改来的，需要修改的地方是文件名和文件内对某些 port 的引用，可以通过用{uut/clk}替代[get_ports {clk}]的方式修改。

Astro 的启动方法是在命令行中输入下面的命令

```
setdt astro
astro_shell&
```

等弹出的窗口的状态栏显示 Ready 时输入 load “run.cmd”即可执行布局布线命令，其中 run.cmd 为布局布线脚本的文件名。执行过程中会出现很多对话框，不要随意操作，等脚本执行完后通过查看 log 来确定脚本的执行状况，Astro 的 log 一般在输入命令的目录下，命名规则为 Astro.log.月_日_时_分。Astro 也有相应的 area 和 timing 报告，生成的文件名一般以模块名为前缀，以.rpt 为后缀。

在 DRC 的时候可能会报很多关于 M6 的错误，可以直接无视，具体的 DRC 和 LVS 会在 Calibre 中进行。

五、 后仿真

工具：Modelsim
输入文件：网表文件（*.vg）、标准延时文件（*.sdf）、标准单元模型文件（smic18.v 或 smic18_neg.v）、Pad 模型文件（SP018W_V1p7.v）、testbench
输出文件：无
目的：检验布局布线后的设计是否能实现所需功能

后仿真与网表仿真类似，也需要指定标准延时文件（*.sdf），不同的地方如下：

①由于 Modelsim 不能识别*.vg 文件，需要将其重命名为*.v 或*.sv；

②由于 PVDD1CAPW 和 PVSS1APW 没有模型，需要将其删掉；

③Pad 模型文件（SP018W_V1p7.v）同样需要包含到 Project 中编译；

④由于顶层文件例化的时候修改了 port，testbench 也应当做相应的修改；

同样需要注意的是要把重名模块移除 Project，否则会因为*.sdf 不匹配而报错。

如果仿真结果不能正确的实现逻辑功能，或有不能忽略的 Error、Warning 则需要重复前面的流程，直到能完成所需功能。

* 仿真工具也可以选用 NC-Verilog，具体用法向数字组同学请教。

数模电路的拼接与验证

完成数字电路设计的所有流程后可以得到两个重要的文件：布局布线后的网表文件（*.vg）和 GDS 文件（*.gds），它们是进行数模电路拼接与验证的关键。后续工作主要包括拼接前数字部分版图的 DRC 与 LVS 和拼接后数模电路版图的 DRC 与 LVS。

一、 数字部分版图的 DRC 与 LVS

工具：Cadence Virtuoso、Calibre、v2lvs

输入文件：数字部分版图 GDS 文件（*.gds）、布局布线后的网表文件（*.vg）、DRC 规则、LVS 规则

输出文件：数字电路的 Spice 网表

目的：检查数字部分版图是否符合 DRC 规则、是否与布局布线后的网表一致

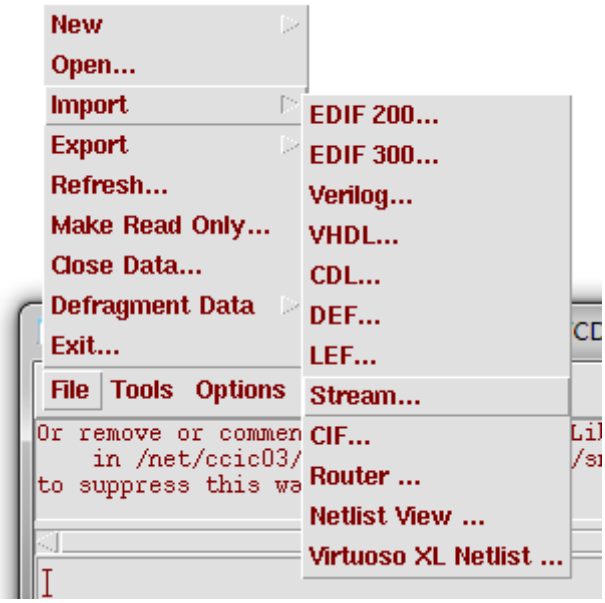
相对于 Astro 中的 DRC 与 LVS，我们更相信 Calibre 给出的结果。但在做 DRC 与 LVS 之前需要先将数字部分版图导入到 Cadence 中，这也是与模拟部分版图拼接的基础。

1. 导入数字版图

①在命令行中输入下面命令可以启动带有 Calibre 插件的 Cadence；

```
setdt ic
setdt calibre
icfb&
```

②在 icfb 中依次点击 File→Import→Stream 可以打开导入 GDS 的对话框；

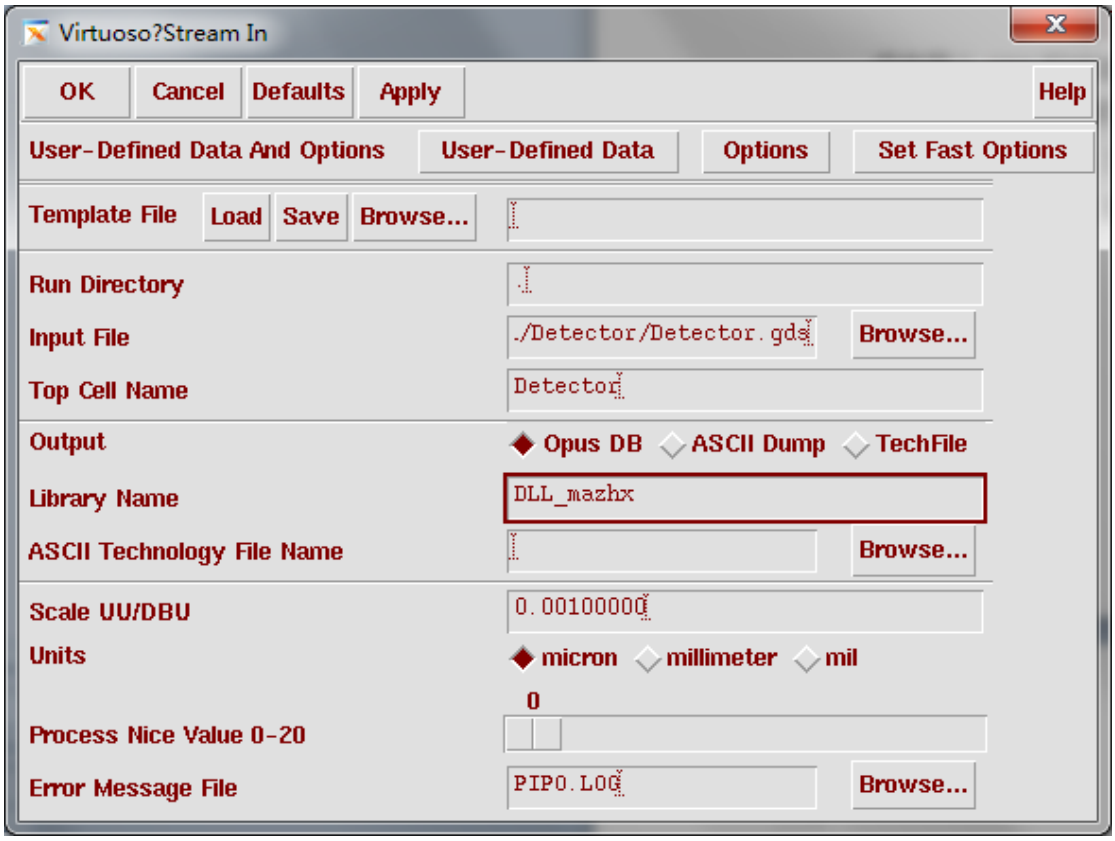


③在导入 GDS 的对话框中需要填写以下内容

Input File：需要导入的 GDS 文件的路径

Top Cell Name：顶层模块的名字，同样也是导入到 Cadence 后 Cell 的名字

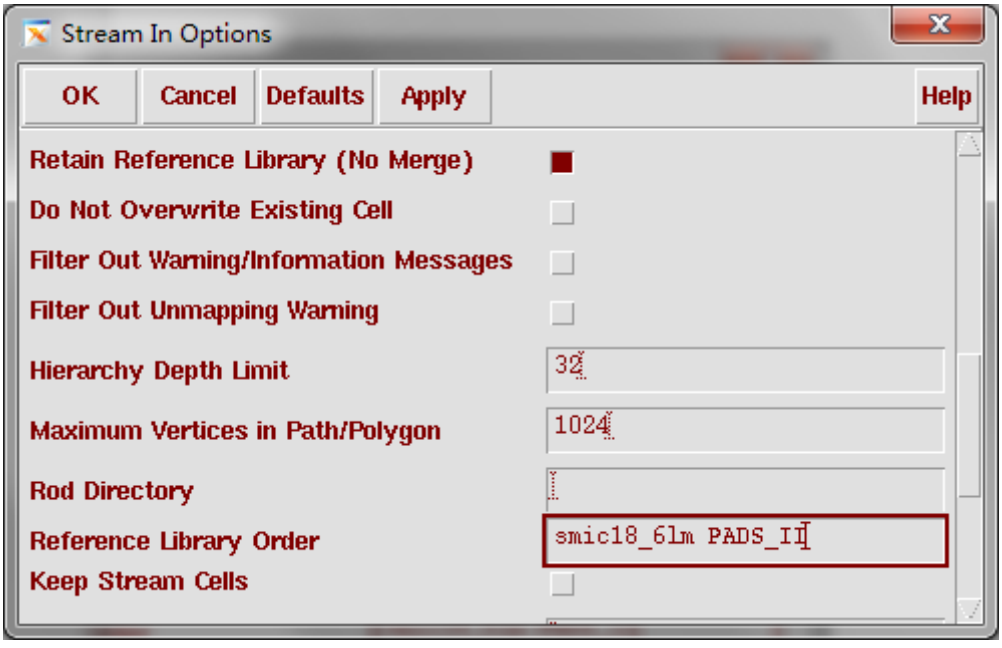
Library Name：准备导入到的库，如果有同 Cell Name 的版图会被覆盖掉



④点击 Options，在弹出的对话框中设置如下

选中 Retain Reference Library (No Merge)选项

Reference Library Order 中填写 smic18_6lm PADS_II，在导入前应保证这两个库都已经被添加到了 Library Path 中。



⑤点击 OK 后等待一会可以看到 GDS 导入的提示，这时打开 Library Manager 在相应的 Library 下面可以看到 Cell 和 Layout。

在导入时过孔都被变为了元件，以$$via 做前缀，放在当前 Library 下面。

打开 Layout 查看生成的版图：如果看不到 pad 环或基本单元需要检查导入的时候 Reference Library 是否设置正确、smic18_6lm 和 PADS_II 库是否已被添加到 Library Path 中；如果看不到 Pin 或 Pin 只有文字没有金属块则需要检查 Astro 脚本导出 GDS 文件时是否选中了 Output Pins 的 As Text 和 As Geometry 选项。

Astro 的 label 默认是打在金属层上的，为了后面的操作建议改成 MxTXT 层（x 代表金属层相应的数字），这个处理可以通过查找/替换完成。

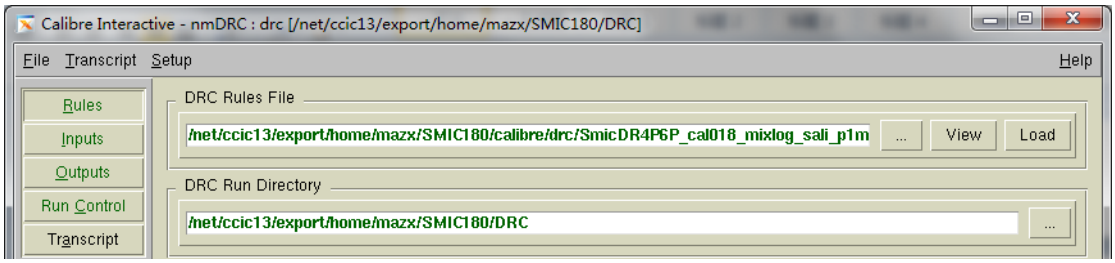
2. 数字版图的 DRC

前面的操作把 GDS 文件导入到了 Cadence 中，这样就可以在 Virtuoso 中修改任何细节，也同样可以做 DRC。

①打开需要做 DRC 的版图，确认 Virtuoso 的菜单栏上有 Calibre 项，这是 Calibre 插件激活的标志，如果没有需要退出 Cadence 按照前面所讲的输入命令；

②点击 Calibre→Run DRC，弹出 DRC 或者 nmDRC 的窗口；

③如果之前没有做过 DRC 的话需要设置 DRC Rule File 的路径和 DRC Run Directory，前者是由 Foundry 提供的工艺相关文件，后者是 Calibre 放置 DRC 生成的临时文件的目录；如果之前做过 DRC 的话应该有保存的 Runset 可以用，前面的设置会自动完成；



④点击左侧的 Inputs，选中 Export from layout viewer 选项；

⑤点击 Run DRC，稍等一会就可以在 RVE 中看到 DRC 结果。

如果 DRC 结果中有一些密度错误，如 MIM_9a 或 PD_XX，则可以被忽略，这种错误将在最后片位拼版时统一解决；如果有其他错误可以双击错误序号，会在 Virtuoso 中看到标注；如果关闭显示 DRC 结果的窗口后想再次打开，可以点击 Run RVE。

3. 数字版图的 LVS

相对于数字版图的 DRC 来说 LVS 的操作更加繁琐一些。LVS 的实质是将从 Layout Viewer 中导出的网表与给定的网表相对照，检查是否匹配。

首先需要将布局布线后生成的网表文件（*.vg）由 Verilog 格式转换成 Spice 格式，其原因是对于 pad 和基本单元的描述是 Spice 格式的。所用到的工具是 v2lvs，其用法如下所示：

v2lvs -v verilog_design_file -o output_spice_file [-l verilog_lib_file] **[-lsp spice_library_file]** [-lsr spice_library_file] [-s spice_library_file] [-s0 groundnet] [-s1 powernet] [-sk] [-p prefix] [-w warning_level] [-a array_delimiters] [-c char1[char2]] [-u unnamed_pin_prefix] [-t svdb_dir] [-addpin pin_name] [-b] [-n] [-i] [-e] [-h] [-cb][-ictrace]

例： `v2lvs -v MyModule.v -o MyModule.netlist -lsp cdl/smic18.cdl -lsp cdl/SP018W_V1p7.sp`

该命令在命令行里输入，如果提示 Command not found 则需要 setdt calibre。

用文本编辑工具打开生成的 Spice 网表，将最后的.GLOBAL 语句删除，再添加下面几句：

```
.GLOBAL VDD!
.GLOBAL VSSD!
.GLOBAL SAVDD
.GLOBAL SAVSS
```

如果使用了数字 pad 还要添加

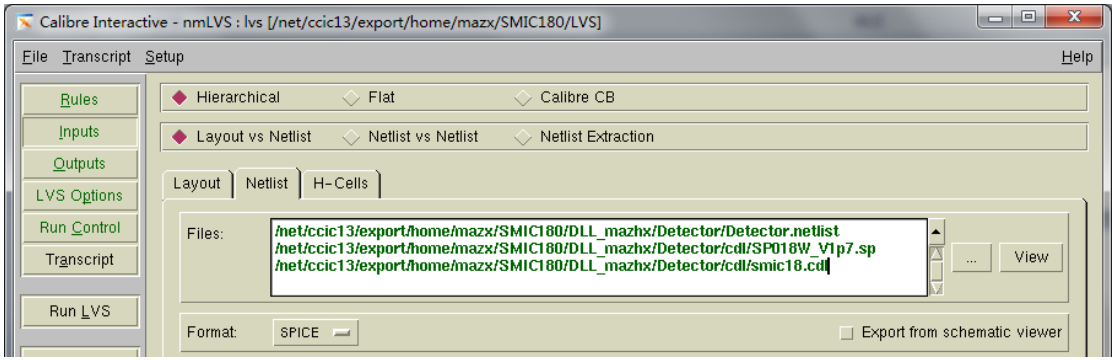
```
.GLOBAL VDD33!
.GLOBAL FP!
```

并在 pad 类型为 PVDD2W 的后面加上 VDD33!=VDD33! FP!=FP!，所有的 pad 后面加上（包括 VDD1 和 VSS3）FP!=FP!。

查找内部接 1'b1 和 1'b0 的网表，将网表中的 VDD 替换为 VDD!，将 VSS 替换为 VSSD!。

打开 Layout，将刚才声明 GLOBAL 的网表都打上 label，否则做 LVS 时会提示 port 数不符。

和 DRC 类似的方法打开 Calibre LVS 的界面，设置 Rule File 和 Run Directory 后点击 Inputs，在 Layout 选项页选中 Export from layout viewer 选项，在 Netlist 选项页清空 Files 中的内容，将刚才修改好的网表和 smic18.cdl、SP018W_V1p7.sp 的路径都添加进去，并取消 Export from schematic viewer 选项。



点击 Run LVS，稍等一会就可以在 RVE 中看到 LVS 结果。

如果 LVS 成功可以看到笑脸，否则会看到若干错误提示，根据提示一一排查解决 LVS 问题。有时候会看到有 ERC 错误，出现错误的位置是某一个电源 pad，据说是误报，可以忽略。

二、 数模电路版图的 DRC 与 LVS

工具：Cadence Virtuoso、Calibre、v2lvs

输入文件：模拟部分版图、模拟部分网表（*.src.net）、数字部分版图 GDS 文件（*.gds）、修改后的网表文件（*.vg）、DRC 规则、LVS 规则

输出文件：数模电路的 Spice 网表文件

目的：检查拼版后整体电路是否符合 DRC 规则、与网表文件是否一致

1. 数模版图的拼接与 DRC

与画模拟版图类似，在 Cadence 中新建一个 Cell，把数字部分版图和模拟部分版图分别加进来，按照事先设计好的位置放置；然后手动的把模拟电路与数字电路的接口之间、模拟电路与 pad 之间用金属线连接起来。为了防止数字电路与模拟电路的衬底相互干扰可以用 M1_NW 将二者隔开。

数模版图的 DRC 与普通的 DRC 完全一样，按照一般流程走即可。下面是 smic18 工艺下常需注意的几种 DRC 规则：

M1-M1	0.23u	M2-M2	0.28u	M3-M3	0.28u
M4-M4	0.28u	M5-M5	-	M6-M6	1.50u
SP-SP	0.44u	SN-SN	0.44u	NW-NW	1.40u
GT-GT	0.25u	AA-AA	0.32u		
Poly-AA	0.10u	N+-N+	0.28u	P+-P+	0.28u
MIM-Mx	2.00u				

2. 数模版图的 LVS

根据 LVS 原理可以知道需要先得到数模电路总的网表才能进行 LVS 的验证，那么可以把模拟电路看做是一个元件，将其 I/O 口抽象出来写成 module 的形式，然后创建一个顶层模块将数字部分、模拟部分、pad 等都例化在一起，用内部连线将它们连接。

按照上述思路，直接修改布局布线后的网表文件（*.vg），将其 I/O 改成只有模拟部分的电源和地，然后在例化数字模块的语句后面加上例化模拟模块的语句。

假设模拟部分电路有两个电源 VDD1、VDD2，一个地 VSS 和若干引脚 pin1、pin2、pin3、pin4，那么网表文件可以写成如下的形式：

```
module 数模版图 Cell 名 ( VDD1, VDD2, VSS );
    // I/O 口声明;
    input VDD1, VDD2, VSS;
    // 内部连线声明;
    wire pin1, pin2, pin3, pin4, ...;
    // 数字模块例化声明;
    数字模块名 例化名 ( ... );
    模拟模块名 例化名 2 (.VDD1(VDD1), .VDD2(VDD2), .VSS(VSS),
    .pin1(pin1), .pin2(pin2), .pin3(pin3), .pin4(pin4));
    // Buffer 例化声明;
    CLKBUFX20 buf_...(.A(...), .Y(...));
    // 数字 pad 例化声明
    PIW     Pad_...(.C(...), .Y(...));
    PO16W   Pad_...(.I(...), .Y(...));
    PVDD1W  Pad_VDD_D(.VDD(1'b1));
    PVDD2W  Pad_VDD33();
    PVSS3W  Pad_VSS_D(.VSS(1'b0));
    // 模拟 pad 例化声明
    PANA1APW    Pad_pin1(.PAD(pin1));
    PANA1APW    Pad_pin2(.PAD(pin2));
    PANA1APW    Pad_pin3(.PAD(pin3));
    PANA1APW    Pad_pin4(.PAD(pin4));
    PVDD1CAPW   Pad_VDD1(.SVDD1CAP(.VDD1));
    PVDD1CAPW   Pad_VDD2(.SVDD1CAP(.VDD2));
```

```
    PVSS1APW    Pad_VSS(.SVSS1AP(VSS));
    // 隔离 pad 例化声明
    PDIODE8W    PAD_DIODE_1();
    PDIODE8W    PAD_DIODE_2();
endmodule
```

最终的网表有以下特征：

①在布局布线后的网表文件（*.vg）的基础上进行修改，只修改顶层模块，其他模块保持不变；

②原顶层模块的模块名修改成拼版后的 Cell 名，I/O 只有模拟电路部分的电源和地；

③元件例化包括数字模块、模拟模块、Buffer 和 pad 四部分；

④内部信号线保证和版图一样的连接，包括 pad 的信号线，除了 ESD pad 和二极管 pad 应该都有信号连接。

总网表中对模拟部分的描述只是例化，还需要模拟电路的网表才能保证电路的完整。模拟部分的版图在之前一定做过 LVS，在 LVS Run Directory 目录下会自动生成一个名为“模拟模块名.src.net”的文件，该文件是从 Schematic Viewer 提取的网表，即模拟电路的网表。

接下来用 v2lvs 工具把模拟部分的网表、标准单元和 pad 的 cdl 都链接起来就可以生成 Spice 格式的总网表了。也就是说在原 v2lvs 命令后面加上“ –lsp 模拟模块名.src.net”，需要注意路径的正确性。

例： `v2lvs -v MyModule.v -o MyModule.netlist -lsp cdl/smic18.cdl -lsp cdl/SP018W_V1p7.sp` **`-lsp 模拟模块名.src.net`**

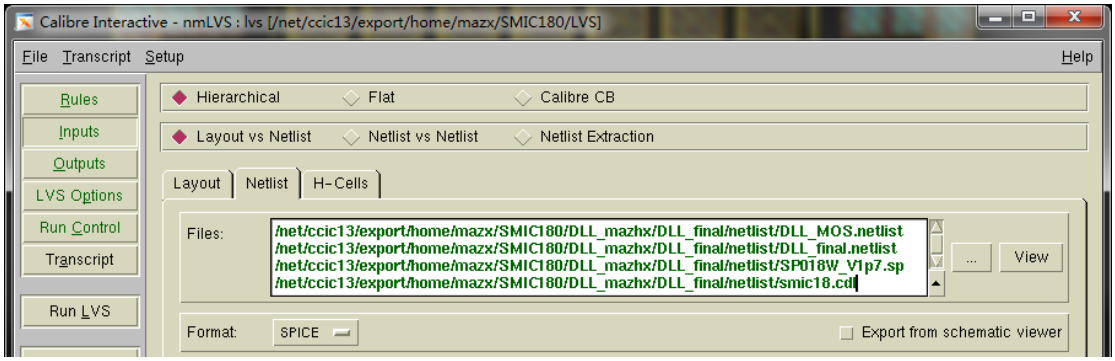
只要总网表没有语法问题，对模拟模块的例化是正确的，v2lvs 对库的引用路径是正确的，就可以看到生成的 Spice 网表；出现错误提示则根据提示信息排查错误，需要慎重检查模拟端口的部分，要和模拟电路的网表对端口的描述（即 Schematic 中的 Port）吻合。

然后用文本编辑器打开该文件，根据前面讲述过的方法对网表做修改。

至此整体电路的网表文件已经生成完毕了，包括四个文件：修改过的 v2lvs 生成的 Spice 网表文件、模拟部分版图 LVS 时生成的*.src.net 文件、标准单元库（smic18.cdl）和 pad 库（SP018W_V1p7.sp），可以将这四个文件复制到一个文件夹下以方便使用。

打开数模版图，确认版图上的 label 有且只有 Spice 网表中用 GLOBAL 语句声明的 Net name 和模拟电路部分的电源、地，被调用模块上的不算，只考虑当前顶层模块上的 label。

打开 Calibre LVS，完成配置后点击 Inputs，选中 Layout 选项页中的 Export from layout viewer，在 Netlist 选项页中将 Files 框中的文字替换为上述四个文件的路径，取消 Export from schematic viewer 选项。



点击 Run LVS，等待片刻就可以在 RVE 中看到 LVS 的结果。

如果 LVS 成功可以看到笑脸  ，否则会看到若干错误提示，根据提示一一排查解决 LVS 问题。同样可以忽略电源 pad 上的 ERC 错误。

至此，整体数模版图的 DRC 与 LVS 已经完成。

三、 部分 LVS 错误及解决方法

这里收录了几种 LVS 错误及解决方法，对实际电路还需要具体情况具体分析。

Q:	点击 Run LVS 后弹出对话框提示 Source primary cell not found in source database
A:	检查是否把顶层模块的网表文件路径添加到 Netlist 的 Files 中或顶层模块的命名是否和数模版图 Cell 的名字一致
Q:	LVS 不显示笑脸，错误数显示为 0
A:	检查 Layout 的顶层 label 数和顶层模块网表的 port 数加用 GLOBAL 语句声明的 Net 数是否一致
Q:	LVS 报的错误特别多，Net Error 和 Instance Error 数都是 50
A:	检查网表文件中电源线的连接、声明是否正确
Q:	虽然模拟部分的电源和地都通过 port 接了出来，版图上也打了 label 但 Extraction Report 仍然提示电源和地错误
A:	Calibre LVS 对电源和地的名字有一定的匹配模式（电源是"VDD" "SAVDD?" "VCC?" "?VDD?" "?VCC?" "?vcc?" "?vdd?"，地是"VSS" "SAVSS?" "gnd" "GND" "?VSS?" "?vss?"），所以命名时要按照这个匹配模式，否则不能被正确识别
Q:	部分管子的衬底连接错误
A:	可能是多个 VSS 情况下衬底没有隔离，可以用 M1_NW 将报错电路围起来做隔离
Q:	模拟电路部分电源 pad 或地 pad 各报两个管子种类的错误
A:	检查电源和地 pad 是否接反
Q:	部分 Net Error，检查 Verilog 网表没有问题
A:	由于 Verilog 中区分大小写，而 Spice 中不区分，如果 Verilog 网表中有用大小写区分不同 Net 的情况可能会报这种错误。最好的解决方法是养成好习惯，尽量避免出现这种问题。

上面的几种 LVS 错误只是冰山一角，更多的错误应该根据实际电路进行分析才能找到解决方法。实在找不到出错原因时可以将 pad、数字模块、模拟模块拆开分别做 LVS，逐步缩小问题范围进而确定问题所在。当然网表文件需要做相应修改，只需将版图中没有的元件的例化语句注释掉即可（Spice 中的注释符是星号*）。

* 感谢李治、曾辉、史大龙、陈泽等人在数模电路版图验证、数字后端等方面给予的帮助。