

用 SpectreVerilog 进行模数混仿，以 Sigma-Delta ADC 为例

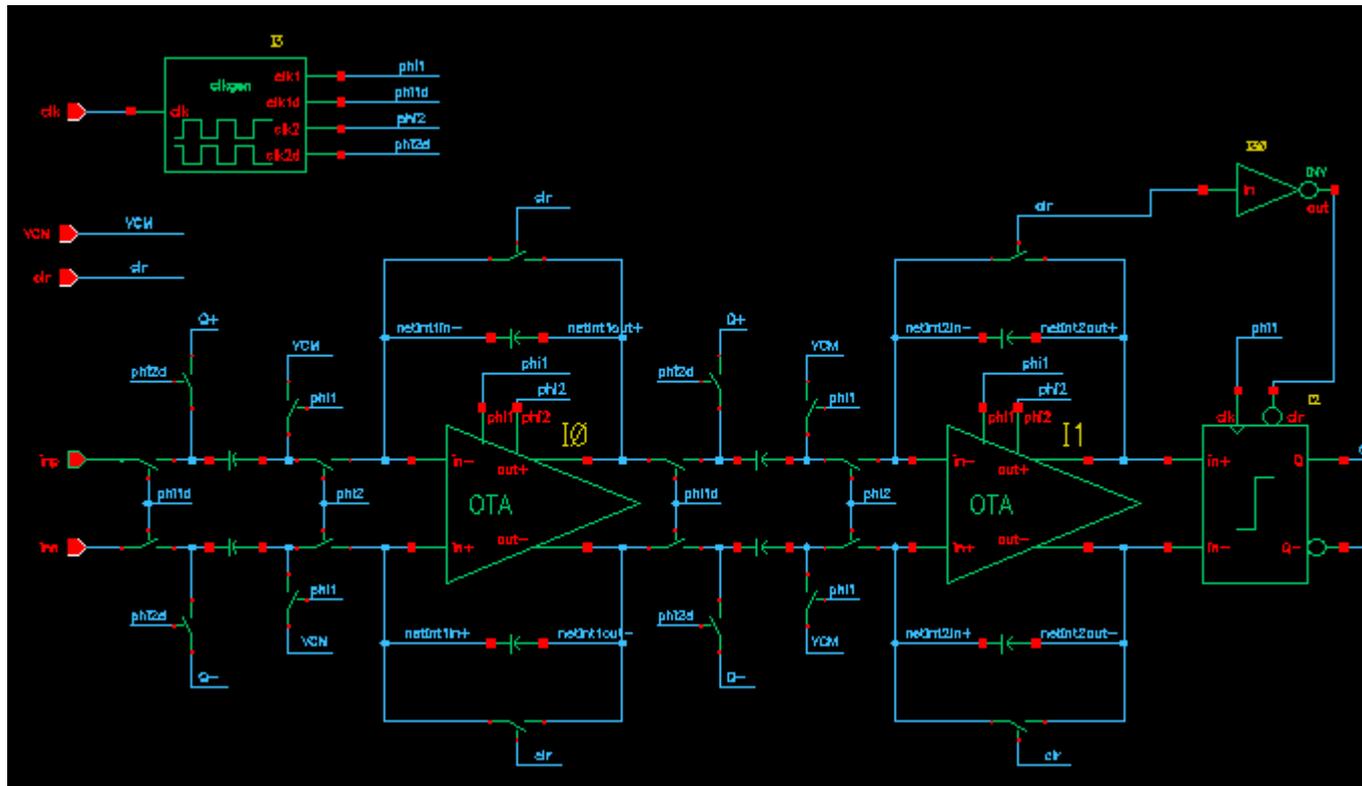
SpectreVerilog 模数混仿，模拟部分用 Spectre，数字部分用 Verilog-XL。所以还需要安装 Cadence LDV 软件，其内含 Verilog-XL 仿真器。

这里以自行设计的二阶全差分 Sigma-Delta ADC 为例，详细介绍用 SpectreVerilog 的仿真过程。所用工艺库为 TSMC 0.18u，电源电压：1.8V。

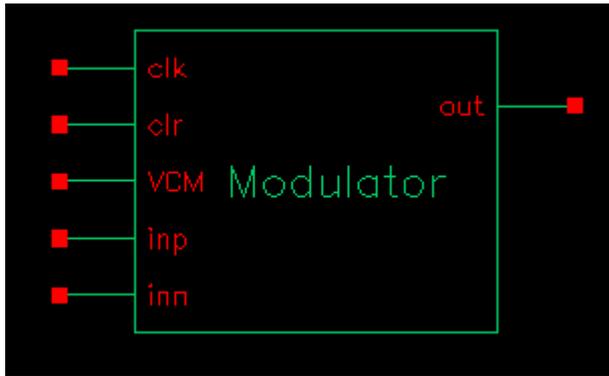
### 1. 准备

Sigma-Delta ADC 分模拟和数字部分两块，其中模拟部分为调制器，数字部分为数字滤波器。如下图。其中 out 为调制器的输出，这里是 1 位 0, 1 数据流。数字滤波器为 Verilog RTL 级代码。

Schematic:



Symbol:



Verilog Code:

```

module DigitalFilter (in2out, out, clk, clr, in);
    output in2out;
    output [^wordsize-1:0] out;
    input clk;
    input clr;
    input in;
    reg in2out;
    wire clk_half1, clk_half2;

    .....
Endmodule

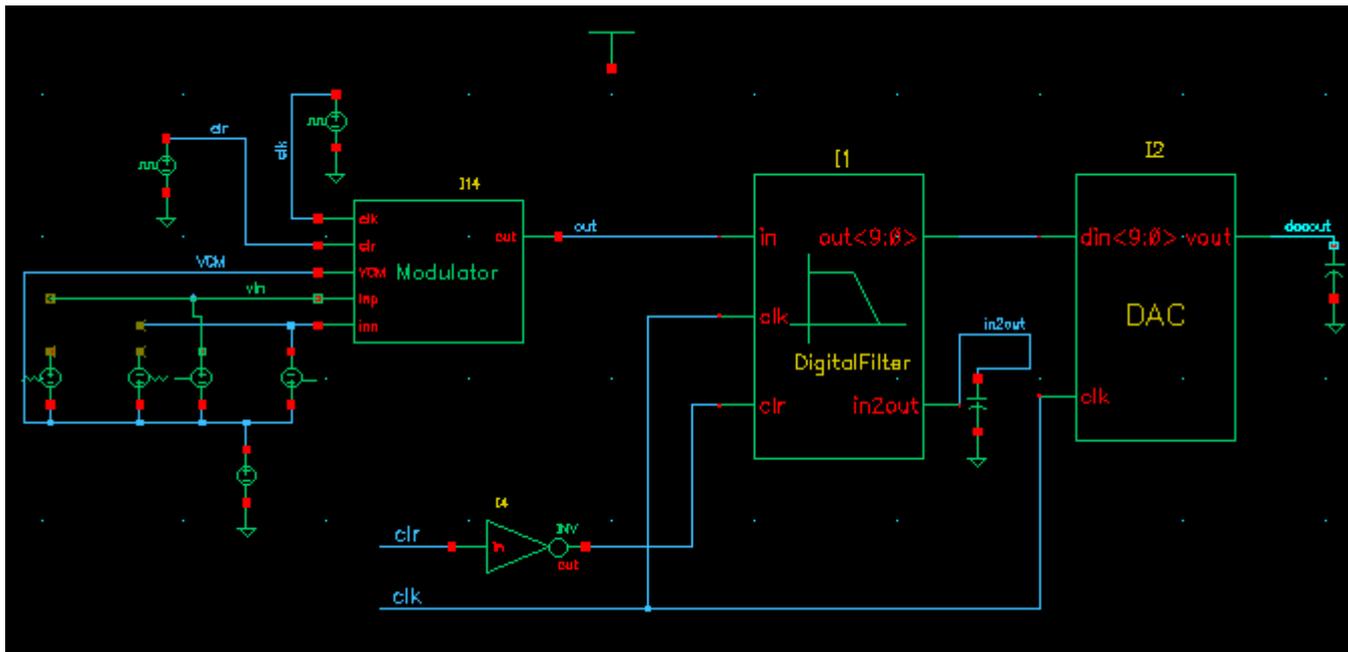
```

同时为了直观的观看输出结果，因此把输出的数字字转化为模拟量，这里用 Verilog-A 做一个理想的 DA 转换器。

因此最好事先用 Spectre 仿真模拟部分，用 ModelSim 或 Verilog-XL 等仿真数字部分。这里假定我们已有：

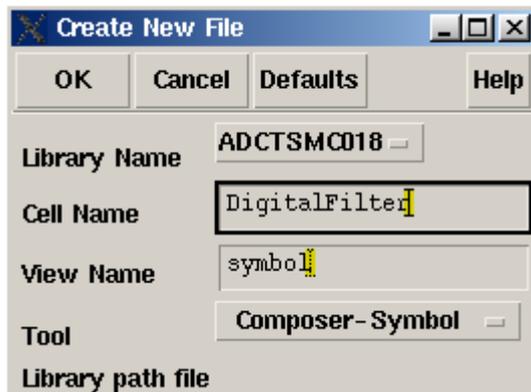
- 1) 模拟部分的原理图(包括 Symbol);
- 2) 数字部分的 Verilog 代码, DigitalFilter.v, 模块名: DigitalFilter(in2out,out,clk,clr,in);
- 3) 数字部分的 TestBench 代码, DigitalFilter\_TB.v, 模块名: DigitalFilter\_TB.

下图为最终的系统图：



## 2. 创建数字模块的 Symbol

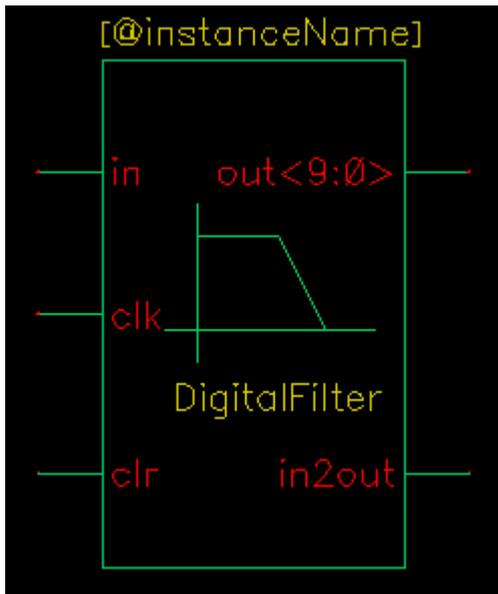
1) 新建一个 Cell, View Name 为 symbol, Tool: Composer-Symbol.



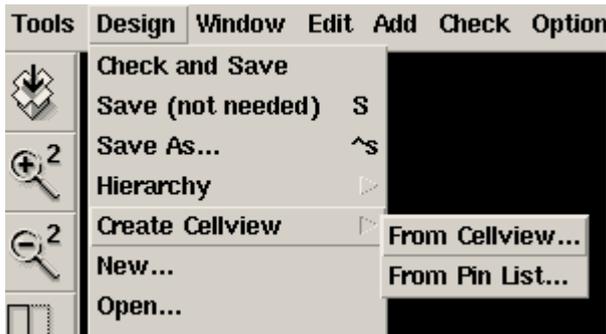
2) 画 Symbol. 简单地, 画一个矩形框, 添加几个 Label, 然后添加 Pin.

3) 添加 Pin. 左边输入, 右边输出. 对于多位的 pin 可以用如 `out<7:0>` 的样式作为 pin 的名字.

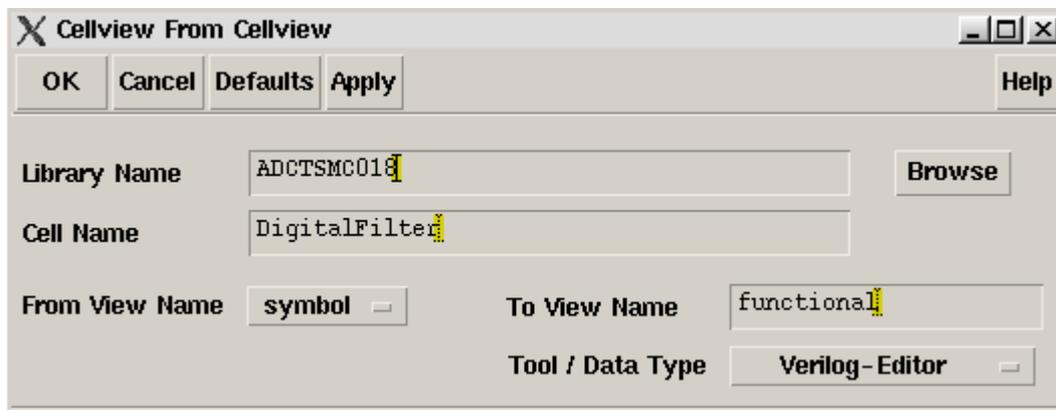
注意: 与模拟部分相连的多位 Pin 最好不要用一个 pin, 而要用多个. 如下图.



4) 创建对应的 Verilog 文件. 在 symbol 编辑器中, Design 菜单->CreatCellView->From CellView.



会弹出的对话框, Tool/Data Type 一栏选择 Verilog-Editor, 则 To View Name 会变为 functional.



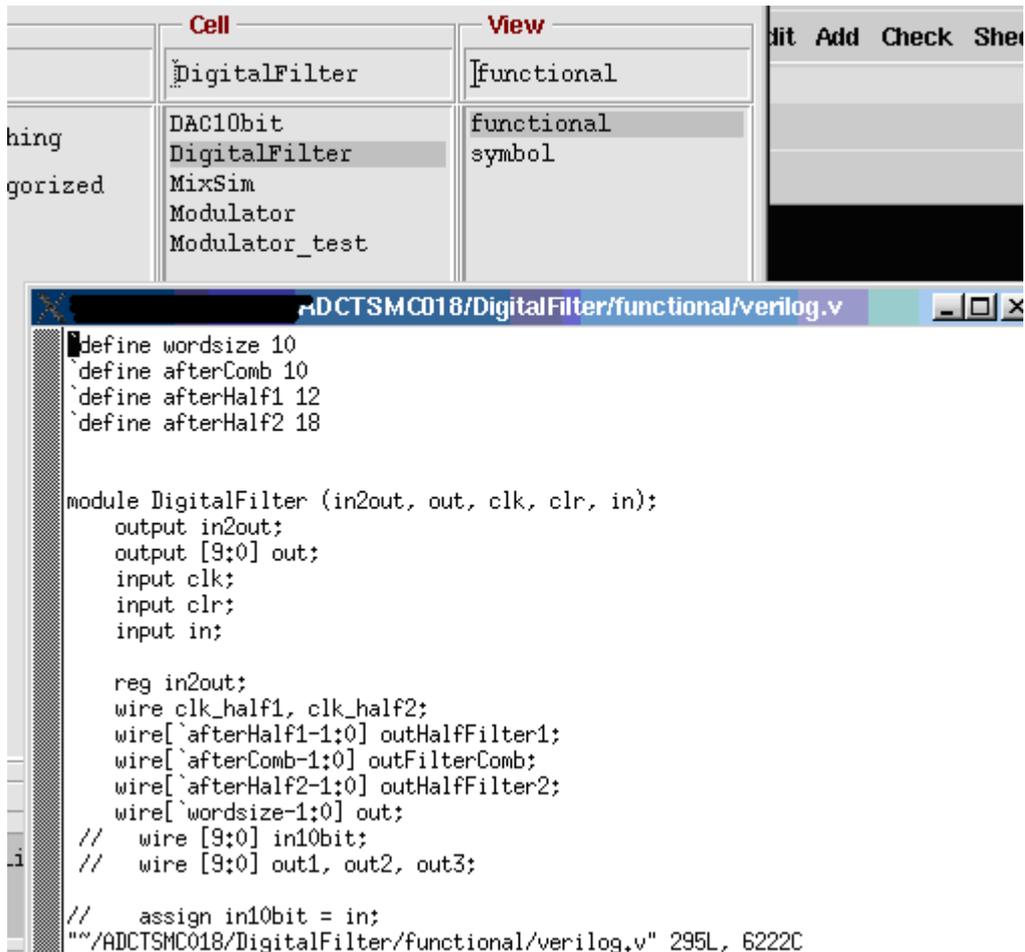
点击 OK, 会弹出错误对话框, 点 No. 自动弹出 VI 编辑器, 可以看到已经生成 Verilog 代码的空壳.

代码文件的路径在 VI 编辑器的标题栏上. 下面要做的就是将我们的数字模块(不是 TestBench)的

代码填进去. 如果不想用 VI 编辑器, 也可以用其他文本编辑器. 复制代码时最好不要动自动生成

的代码. 经测试, 所有代码最好放在一个文件中.

这一步之后, 数字部分就会有 functional 和 symbol 两个 View.

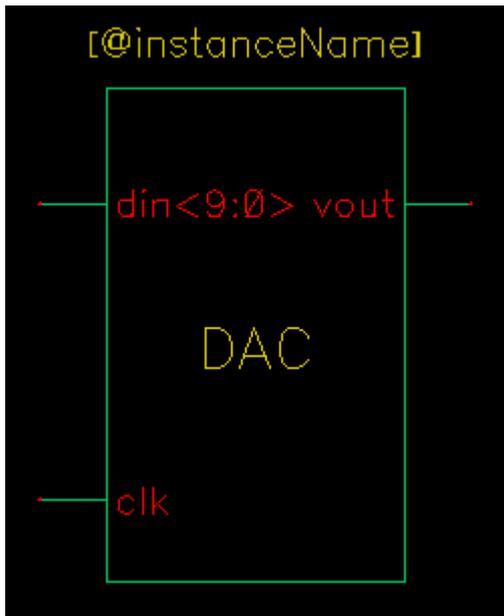


这样基于 Verilog 代码的 Symbol 就创建完了.

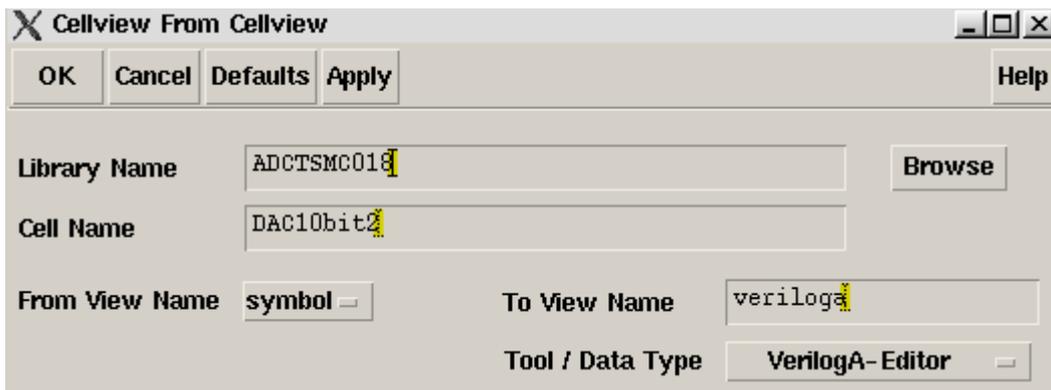
### 3. 创建理想 DAC

这里用 Cadence 自动生成的理想 DAC.

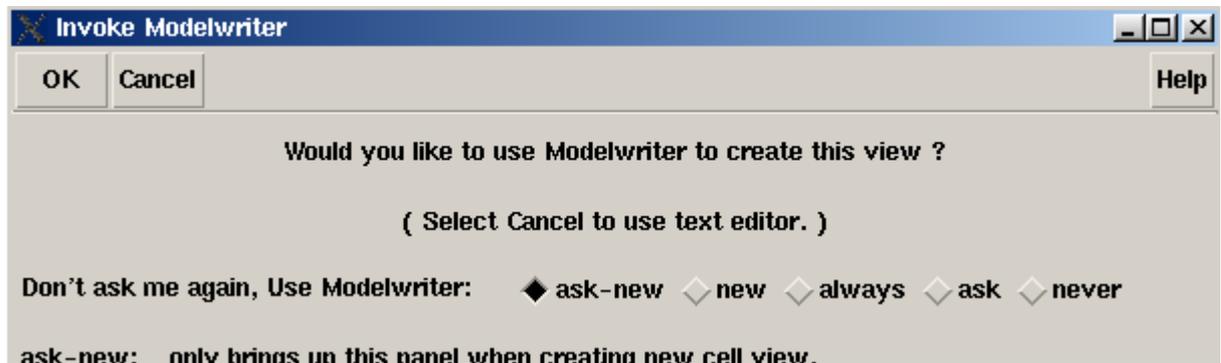
同步骤 2, 首先创建 Symbol, 如下图:



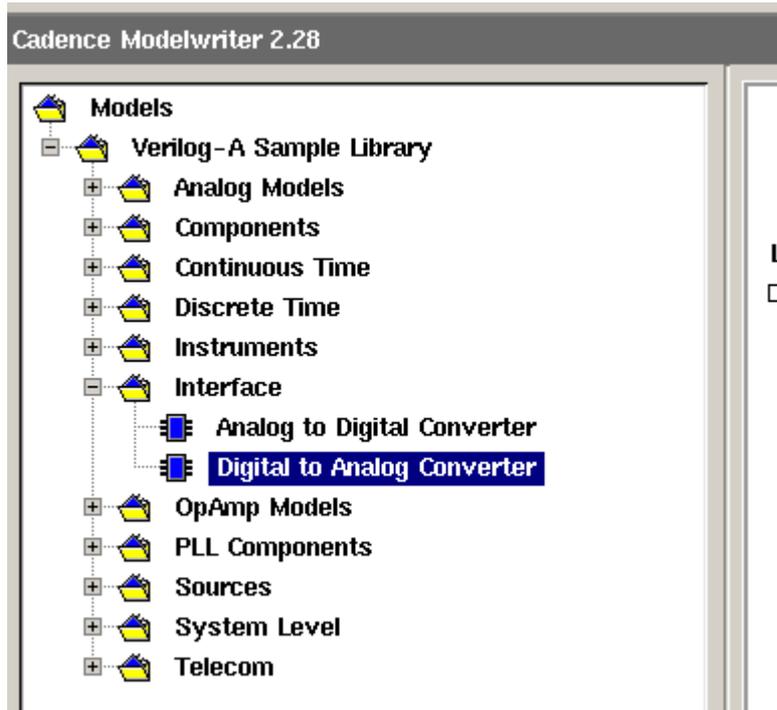
类似步骤 2，Design 菜单->CreatCellView->From CellView. Type 选择 VerilogA-Editor



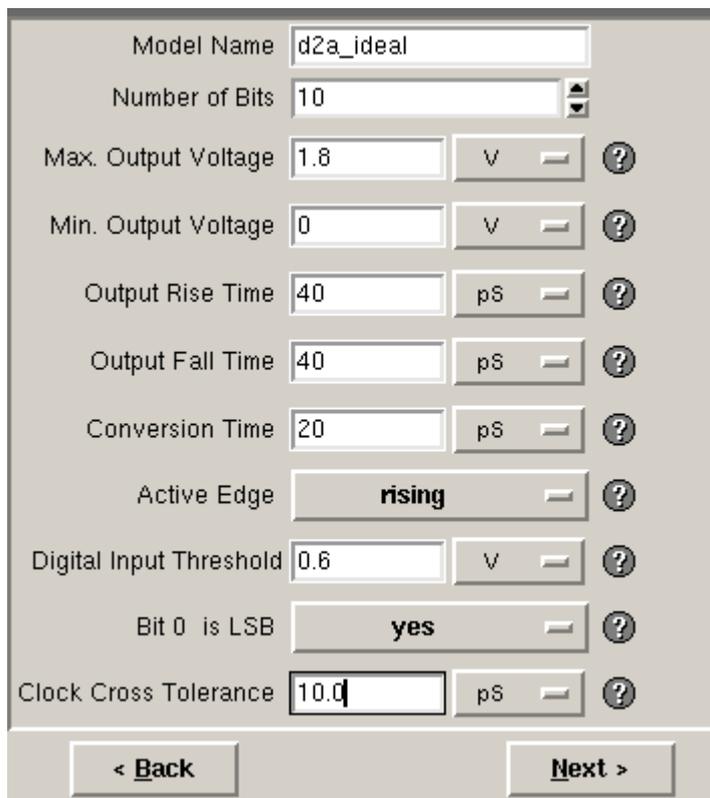
OK，弹出对话框，可以直接按 OK，默认设置。



弹出新的对话框，选择 Digital to Analog Converter, 然后 Next。



设置位数，电压范围等参数。如下图：



Next 之后，会出现 VerilogA 代码，再 Save 和 Finish。完毕。

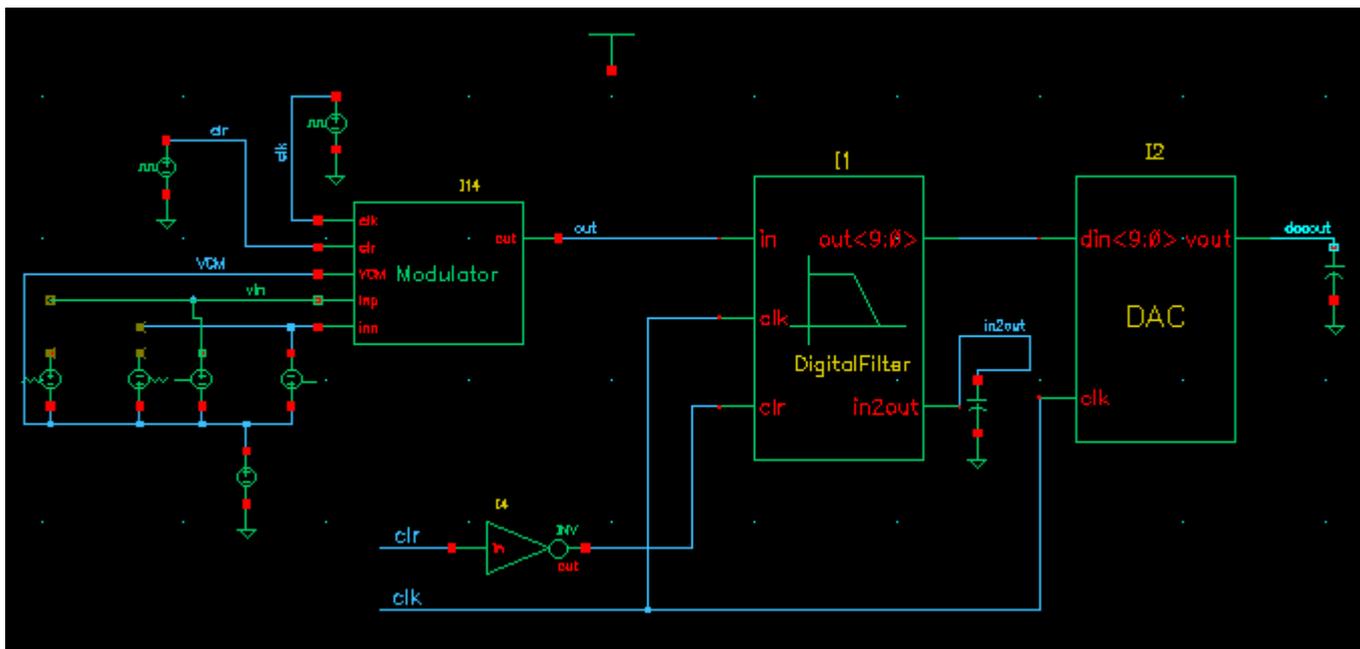
#### 4. 创建顶层原理图.

1) 新建步骤 1 中所说的顶层 Schematic 视图, 这里名为 MixSim, 把模拟的 Symbol 和上面建立的

数字块 Symbol 放到新建的原理图中, 并连好线. 模拟块输入端的激励源都要加上.

2) 添加数字块的 Pin. 把数字块中除了和模拟块相连的输入端添加 Pin. 这里的数字块输入端

信号与模拟块共用, 因此没有输入 Pin.

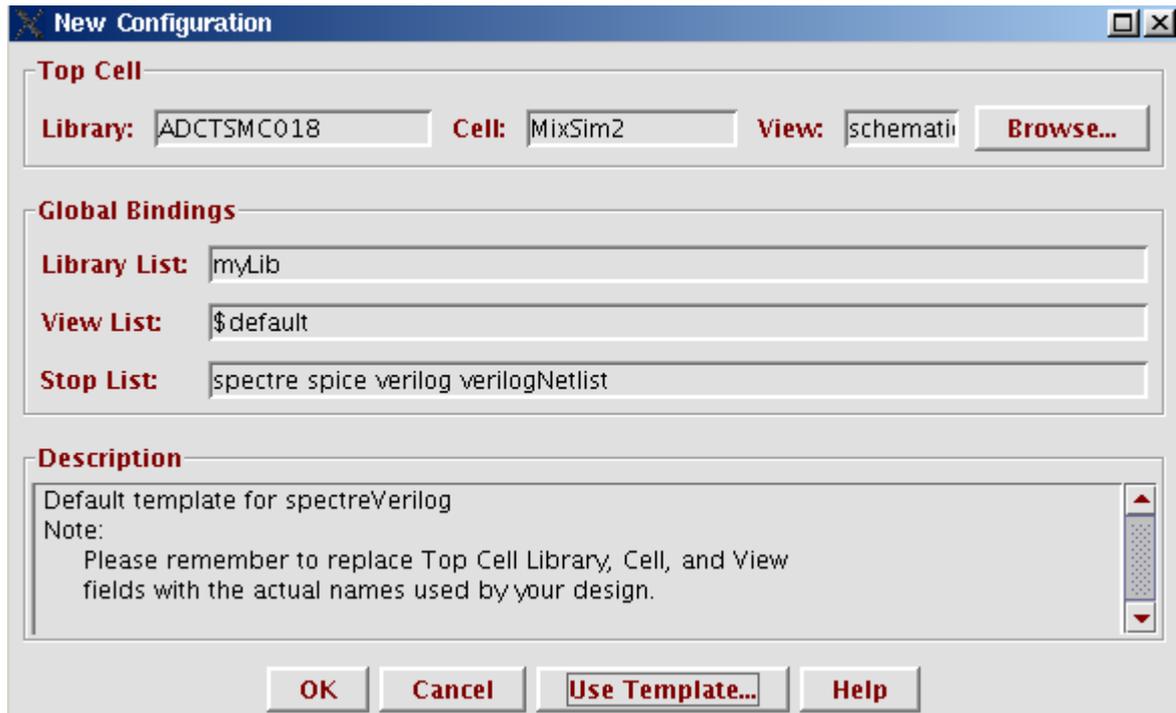


#### 5. 创建 config 视图

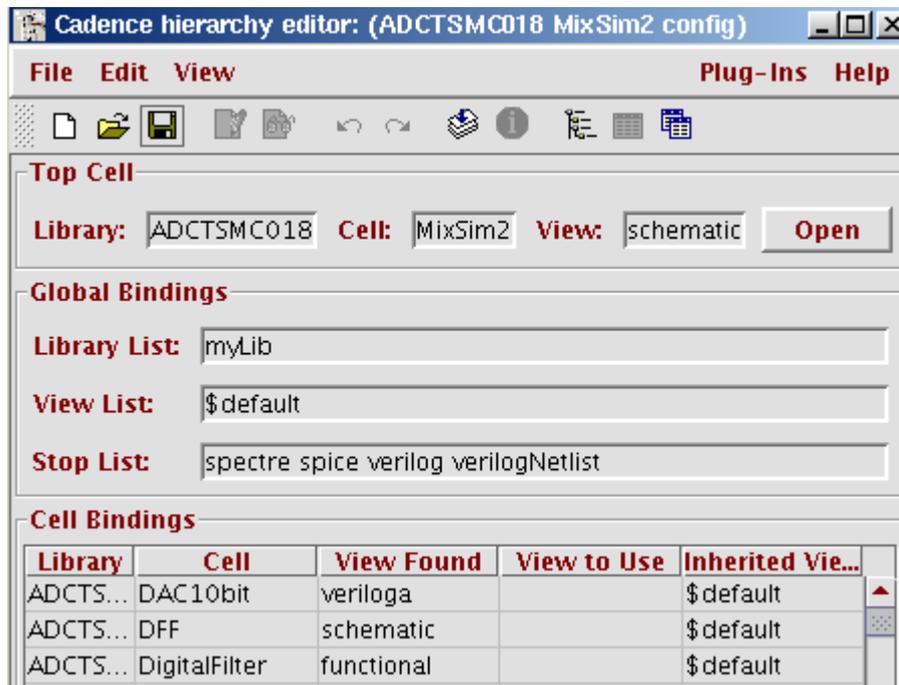
1) 新建 Cell, 名字和步骤 4 中的原理图名字一致. Tool 选择 Hierarchy-Editor, 则 View Name 会自动为 config. OK,



2) 在新出现的 New Configuration 窗口中 View: 文本框为空, 填入: schematic, 注意大小写. 点击下方的 Use Template..., -> Name 一栏选择: spectreVerilog, OK,



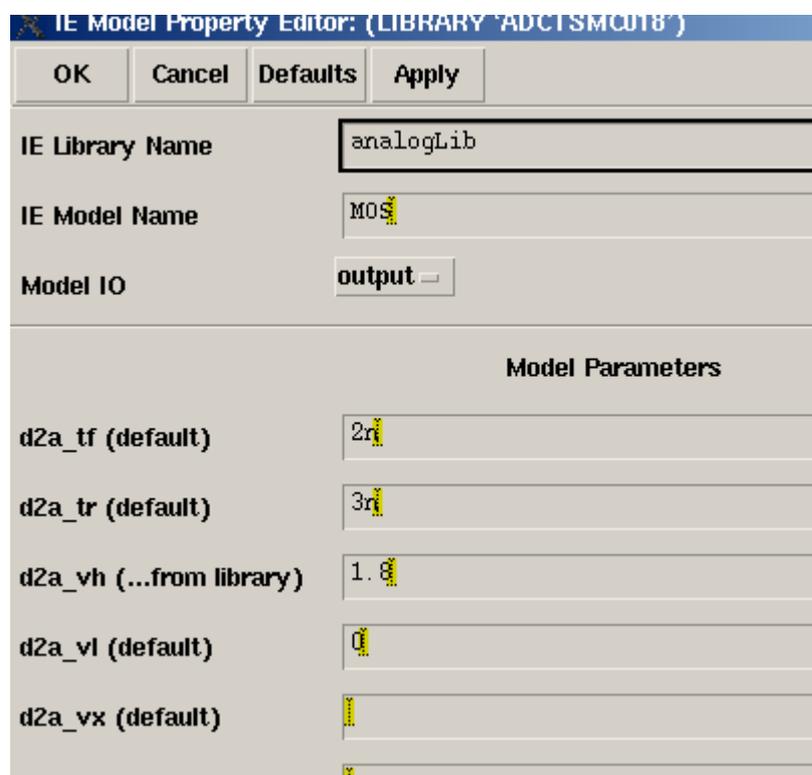
3) 再 OK 关闭 New Configuration. 点击 Cadence hierarchy editor 窗口的保存按钮, 关闭退出.



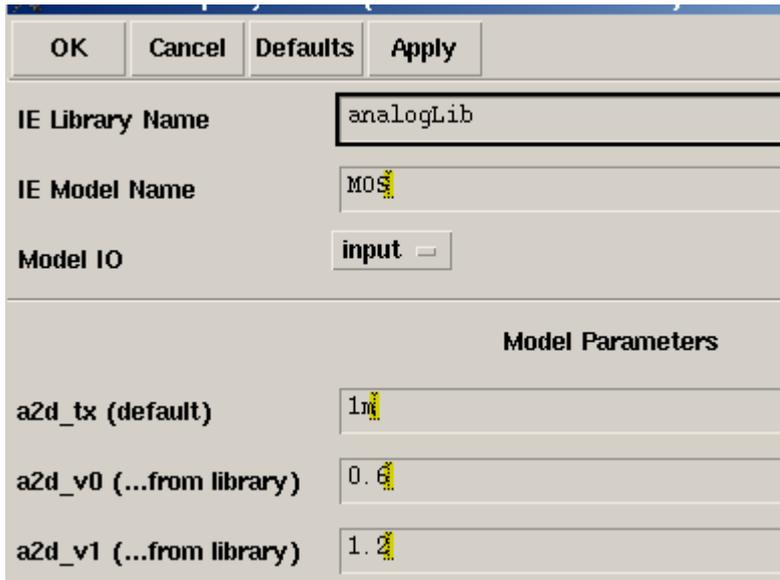
4) 这时 MixSim 有两个 View: config, schematic. 打开 config, 会提示打开模式, 直接 OK, 则原理图会被打开, 并可以进行修改。

6. 建立仿真环境

- 1) 打开 config 视图, 菜单 Tools->Analog Environment, 打开 ADE.
- 2) ADE, 菜单 Setup->simulator, Simulator 选择 spectreVerilog, OK.
- 3) ADE, 菜单 Setup->Stimuli->Analog, 弹出 Setup 窗口, 选中 Global Sources, 把电源电压填上, 这一步和 spectre 一样, 另外别忘了添加模型库. 不再赘述.
- 4) 添加数字块的测试向量. ADE, 菜单 Setup->Stimuli->Digital, 弹出 VI 编辑窗口, 窗口的标题栏和信息栏都有测试向量的文件路径, 可以直接修改它. 系统自动生成了一个 initial 块, 这个 initial 块不能改动. 现在就可以把 Testbench 中的代码复制到这个文件中了. 注意: `timescale 不要复制, 模块名和端口定义不要复制, 模块实例引用不要复制. 也就是只能复制 initial 块和 always 块以及 parameter 语句. 同时注意到, 系统生成的的 initial 块中把输入信号初始化为零, 因此复制进来的语句一定要避免在 0 时刻给这些信号赋值. 本例数字块的输入与模拟块共用, 因此不用添加代码.
- 5) 打开 Config 视图, 菜单 Tools->Mix Signal Opts., 会新增加一个 Mixed-Signal 菜单栏. 菜单 Mixed-Signal -> Interface Elements -> Library... 主要修改两个参数 d2a\_vh, d2a\_vl, 他们是输出数字信号的 0, 1 电压



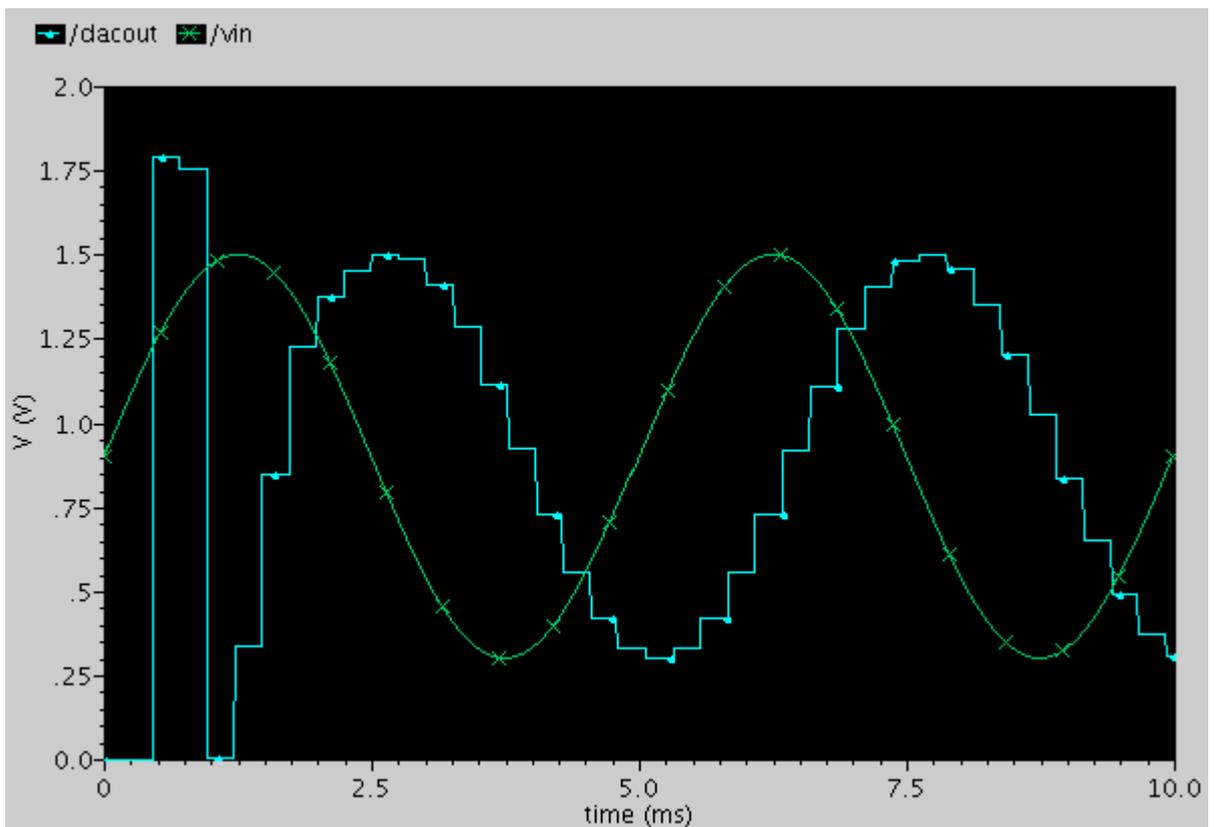
把 Model IO 选为 Input, 修改 a2d\_v0, a2d\_v1, 他们是模拟转数字的阈值电压, 这里为 0.6V 和 1.2V



## 7. 仿真.

仿真过程与 Spectre 一样，设置瞬态仿真时间，输出节点电压。

这是 ADC 仿真结果图:



由于搭建整个仿真环境比较复杂，因此经常会出错，需要耐心调试，尤其是 Verilog 代码错误，

仿真器只说有错，但不知道错在何处，很不爽。

本教程仅适用于初学者，同时请高手指点一二，错误和走弯路都在所难免，谢谢。