

## SPICE网表至VERILOG栅极转换器

# CatalystAD

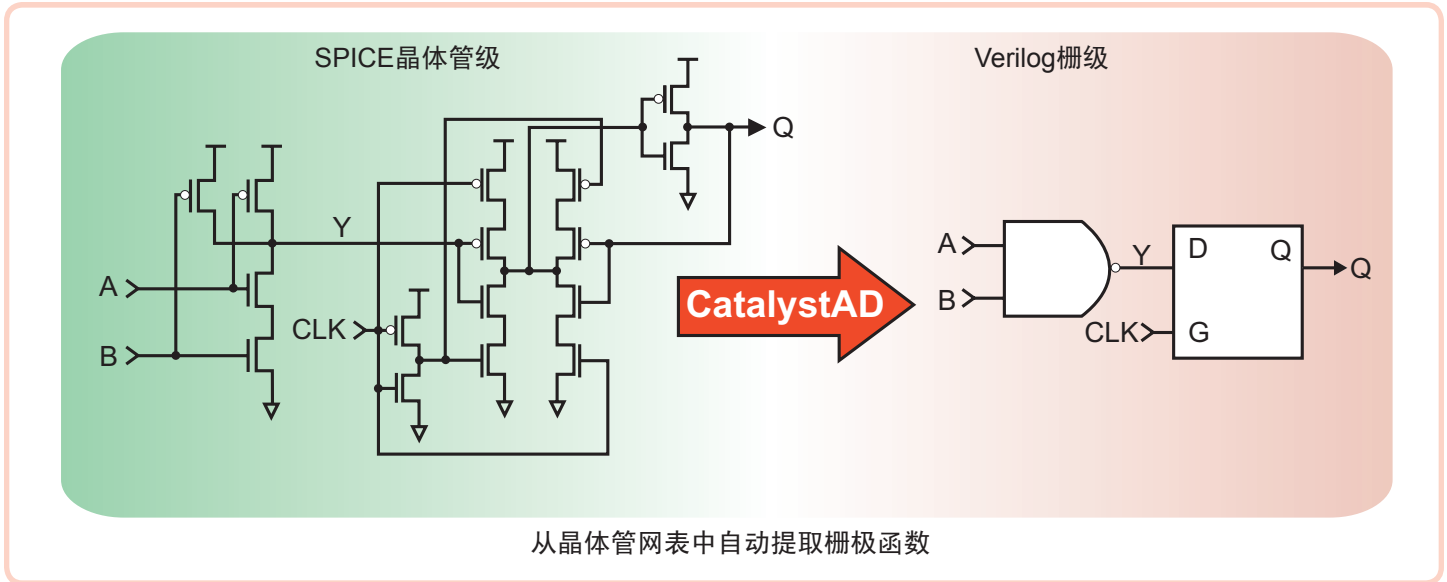
CatalystAD是将晶体管设计转换至verilog栅极网表的最佳工具，它应用于微处理器、DSP、图形和高速通讯领域。



- 可将晶体管网表自动生成Verilog栅极网表和模型
- 是反向工程原有硬IP (hard IP) 和定制逻辑设计的理想之选，以供设计的重复使用和技术迁移
- 支持HSPICE™/SPECTRE™和DSPF的分层式或扁平式网表
- 可处理所有类型的CMOS/SOI设计风格 (标准单元、定制、静态、动态、复合、序列、多米诺、有底脚的、无底脚的、自定时、充电后、共发共基、DCVS、传输晶体管、桶形移位器、交叉开关结构、m-of-n逻辑树等)
- 可控制24个或更多输入的宽扇入传输栅极、潜路径和包括成百上千并行路径的输出路径深度的的正确建模
- CatalystAD联合AccuCore提供了一套完整的验证和时序建模方案

## 特征

- 简便处理带有上百万个晶体管的网表
- 自动提取全定制和IP硬核块函数
- 生成栅级仿真和综合verilog网表及模型
- 支持HSPICE™/SPECTRE™和DSPF分层式或扁平式网表
- 简便使用批模式Tcl脚本和配置文件接口



## 工具流程步骤

1. 读取SPICE/DSPF网表(若必要则展平), 保留模块、端口、网路名和主线完整层次路径信息。
2. 从非静态元件如时钟分配器、时钟加倍器、相位移动器以及PLL中生成时钟, 通过布尔分析执行时钟传播, 并识别后来的时钟节点(它们通过特定节点来终止时钟传播)。
3. 通过黑盒命名用户定义的样式, 从逻辑电路中隔离出模拟电路
4. 自动识别锁存器和触发器结构及其变体
5. 将设计分区成单元。
6. 基于函数提取和单元分类, 执行自动算法和样式。
7. 配合可选黑盒成分, 生成结构verilog单元网表
8. 生成栅级verilog模型

## 输入

- 带有可选RC信息的分层式或扁平式 SPICE/DSPF网表
- 带有命令和变量设置、引脚信息和文件名的.cfg配置文件
- 基本的.tcl运行脚本文件

## 输出

- 结构式Verilog单元网表
- Verilog栅极模型库
- Verilog模板库

## 输入文件

Example and.cfg input file:

```
inputs a b c
outputs out
powers vdd
grounds gnd
TOP_VLOG_MODULE and
TOP_SPICE_SUBCKT and
IN_FILE_NAME and.cir
MOSFET_TYPE p pmos
MOSFET_TYPE n nmos
```

Example and.tcl input file:

```
gen_model and.cfg
```

Run-time command:

```
catalystad and.tcl |& tee and.log
```

## 输出文件

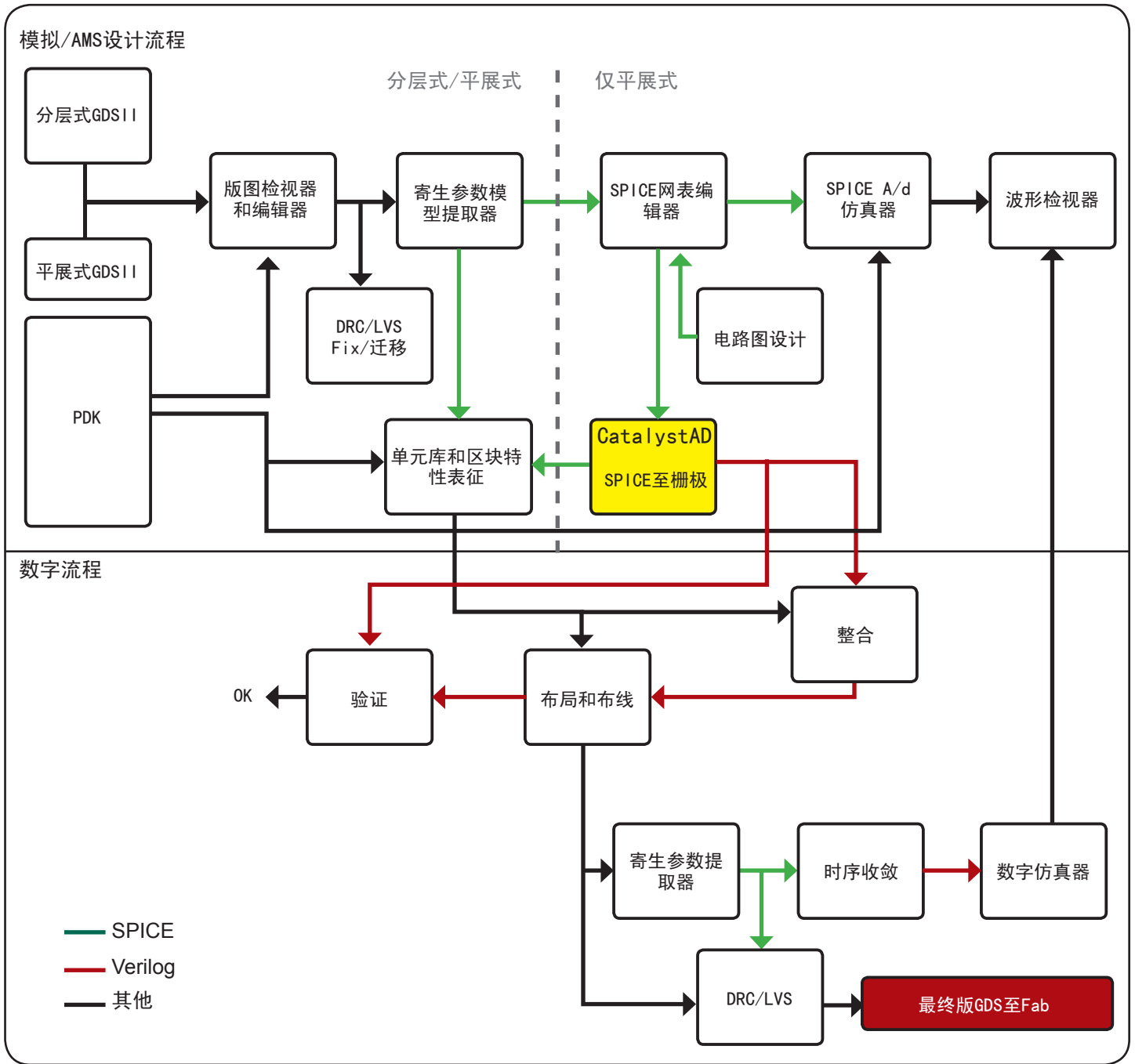
Example and.v netlist output file:

```
module and( out , a , b , c );
output out ;
input a , b , c ;
supply1 vdd ;
supply0 gnd ;
wire z;
and_dc_2 i_and_dc_2( .z(z) , .a(a) , .b(b) , .c(c) );
and_dc_3 i_and_dc_3( .out(out) , .z(z) );
endmodule
```

Example and\_vlg.lib model output file (partial):

```
`include "template.v"
`celldefine
module and_dc_2( z , \a<1> , \a<0> );
output z ;
input \a<1> , \a<0> ;
// gate type static for z
wire net_z_1_0 ;
wire z_out_0 ;
wire net_z_1_1 ;
wire z_out_1 ;
buf ( net_z_1_0 , \a<0> );
mux mux_inst_z_0_0 ( z_out_0 , \a<1> , 1'b0 , net_z_1_0 );
not ( net_z_1_1 , \a<0> );
mux mux_inst_z_0_1 ( z_out_1 , \a<1> , 1'b1 , net_z_1_1 );
// inverter for 0 term
wire z_out_0_bar ;
not ( z_out_0_bar , z_out_0 );
// output driver
and ( z , z_out_0_bar , z_out_1 );
endmodule
`endcelldefine
```

# 设计中心工具流程



# SILVACO

新加坡

**Silvaco Singapore Pte Ltd**

77 Science Park Drive, CINTECH III #03-10  
Singapore Science Park I, Singapore 118256

Tel: +65-6872 3674

Fax: +65-6872 2497

Email: sgsales@silvaco.com

WWW.SILVACO.COM.CN

Rev. 010609\_03