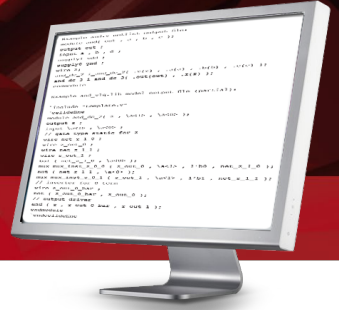


# Catalyst AD

SPICEネットリスト / Verilogゲート・コンバータ



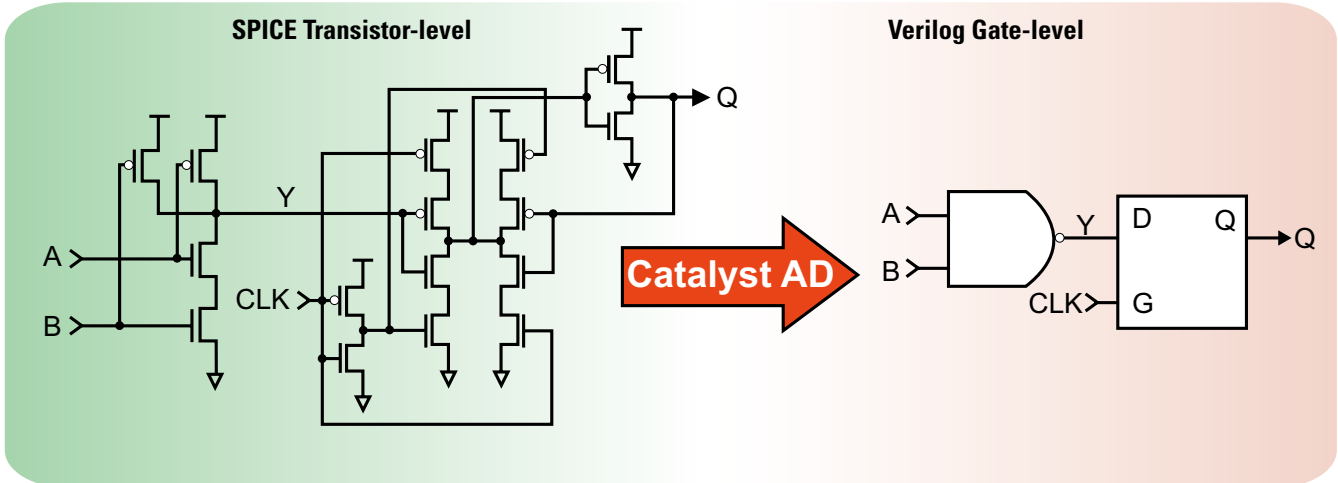
**Catalyst AD** は、トランジスタ・レベルの記述を Verilog ゲート・レベルに変換するツールとして最適です。マイクロプロセッサ、DSP、グラフィック、高速通信分野などの用途に対応しています。

- トランジスタ・レベルのネットリストから、Verilog ゲート・レベルのネットリストおよびモデルを生成するための、自動ソリューションを提供
- ハード IP の設計資産およびカスタム・ロジックの、デザイン再利用 / マイグレーションのための、リバース・エンジニアリングに理想的
- HSPICE™ / SPECTRE™ および DSPF 階層 / フラット・ネットリストに対応
- あらゆる CMOS/SOI デザイン・スタイルに対応 (スタンダード・セル、カスタム、スタティック、ダイナミック、コンビネーション、シーケンス、ドミノ、フット付、フットレス、セルフ・タイムド、ポスト・チャージ、カスコード、DCVS、パス・トランジスタ、バレル・シフタ、クロスバー・スイッチ構造、m-n ロジック・ツリーなどに対応)
- 24 入力以上の幅広いファンインを持つパスゲート、スネーク・パス、数十万もの並列パスを含む出力パスの深さのモデリングを適切に制御
- **Catalyst AD** と **AccuCore** を共に用いることで、完全な検証およびタイミング・モデリングのソリューションを提供

**SILVACO**

## 特徴

- ・ 数百万ものトランジスタにより構成されるネットリストを容易に処理
- ・ フル・カスタムおよびハード IP ブロックに対してファンクションを自動抽出
- ・ ゲート・レベル・シミュレーション用の Verilog ネットリスト / モデルを生成
- ・ HSPICE / SPECTRE および DSPF 階層 / フラット・ネットリストに対応
- ・ 使い易いバッチ・モード Tcl スクリプトおよび config ファイル・インタフェース



トランジスタ・レベルのネットリストから、ゲート・レベルのファンクションを自動抽出

## ツール・フロー手順

1. 完全な階層パス情報を含み、モジュール / ポート / ネットの名前およびバスを保持しながら、平坦化した SPICE/DSPF ネットリストを読み込む。
2. クロック分周器、クロック通倍器、フェーズ・シフタ、PLL などの非静的素子に定義されたクロックを、ブール代数解析により回路内に伝播させる。
3. アナログ回路を、ブラック・ボックス・コマンドで定義したユーザ定義のパターンにより、論理回路から切り分ける。
4. ラッチおよびフリップ・フロップの構造やそのバリエーションを自動認識する。
5. デザインをセルに分割する。
6. アルゴリズムとパターンに基づいたファンクション抽出およびセル分類を自動的に実行する。
7. Verilog セル・ネットリストを生成する。(素子のブラック・ボックス化オプションあり)
8. ゲート・レベルの Verilog モデルを生成する。

## 入力

- ・ オプションの RC 情報を含む、階層またはフラット SPICE/DSPF ネットリスト
- ・ コマンド、変数設定、ピン情報、ファイル名を含む、コンフィグレーション・ファイル (.cfg)
- ・ 基本のランタイム・スクリプト・ファイル (.tcl)

## 出力

- ・ Verilog セル・ネットリスト
- ・ ゲート・レベルの Verilog モデル・ライブラリ
- ・ Verilog テンプレート・ライブラリ

### 入力ファイル例：

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
```

### 出力ファイル例：

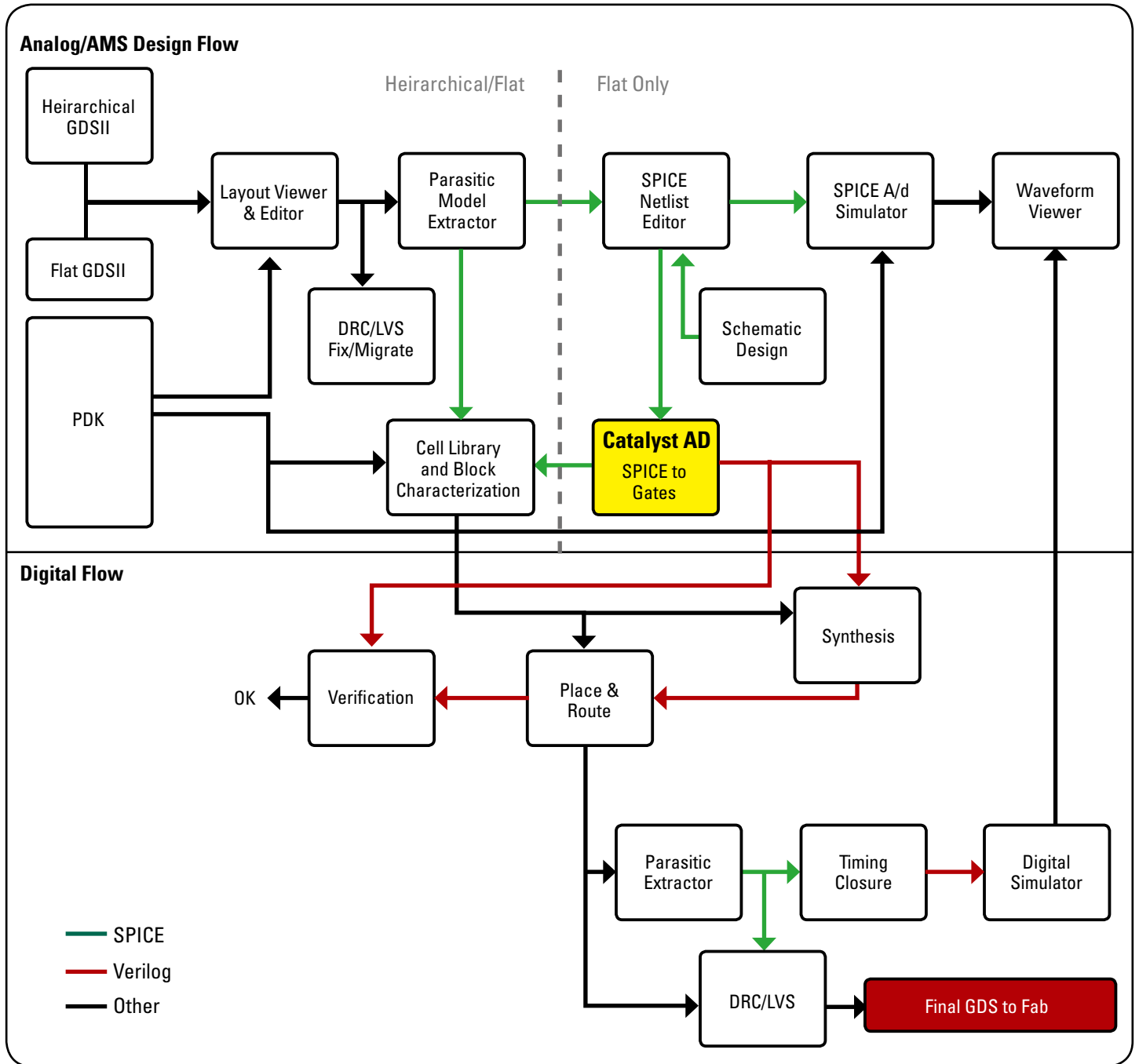
Example and.v netlist output file:

```
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
```

# デザイン・センター・ツール・フロー



**SILVACO**

株式会社 シルバコ・ジャパン  
www.silvaco.co.jp

お問い合わせ : [jpsales@silvaco.com](mailto:jpsales@silvaco.com)

**横浜本社**

〒 220-8136  
 神奈川県横浜市西区みなとみらい 2-2-1  
 横浜ランドマークタワー 36F  
 TEL : 045-640-6188 FAX : 045-640-6181

**京都オフィス**

〒 604-8152  
 京都市中京区烏丸通蛸薬師下ル手洗水町 651-1  
 第14長谷ビル 9F  
 TEL : 075-229-8207 FAX : 075-229-8208

