Mixed-Signal IC Design Kit
Training Manual

Chip Implementation Center

許志賢
chhsu@cic.edu.tw
(03)5773693 ext 147

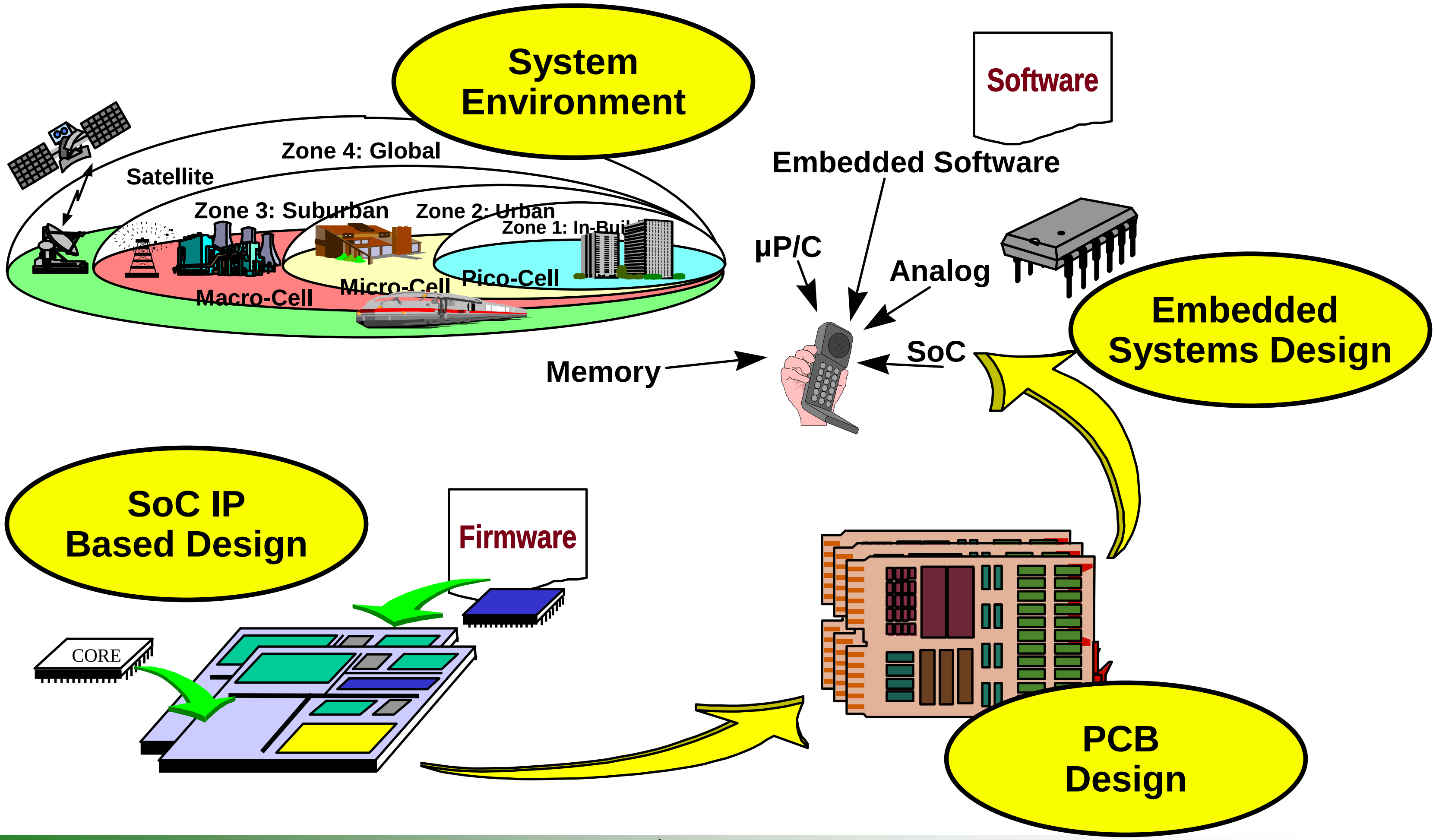
Agenda

- ❑ Day 1
  - ➢ Introduction of Mixed-Signal Simulation
  - ➢ Using Analog Artist Environment for Mixed-Signal Design
- ❑ Day 2
  - ➢ Layout Integration for Mixed-Signal Design
  - ➢ Verification & Post-Layout Simulation for Mixed-Signal Design

Introduction of Mixed-Signal Simulation

Why Mixed-Signal Simulation?

What’s in a System?



System Environment

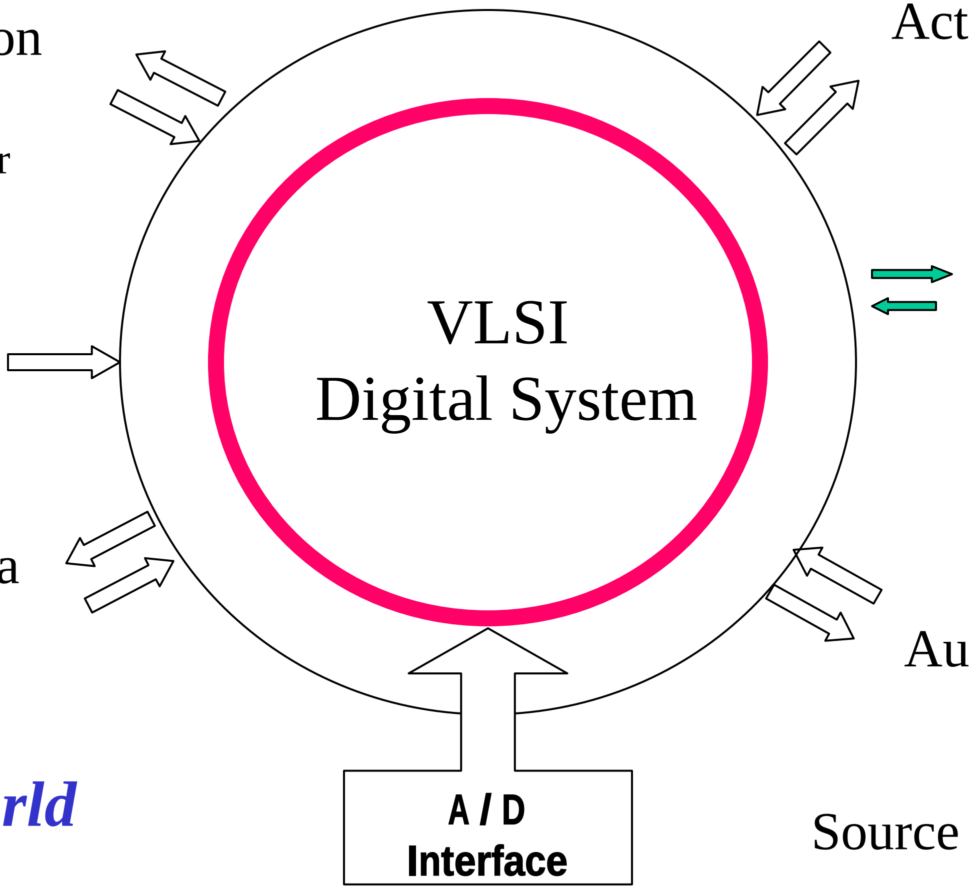
Embedded Systems Design

SoC IP Based Design

PCB Design

System in the Real World

Analog is the Real



VLSI
Digital System

A / D
Interface

Transmission
Media

Cable,fiber
antenna

Power
Source

Storage Media

Disks,
Tapes

Sensor
Actuators

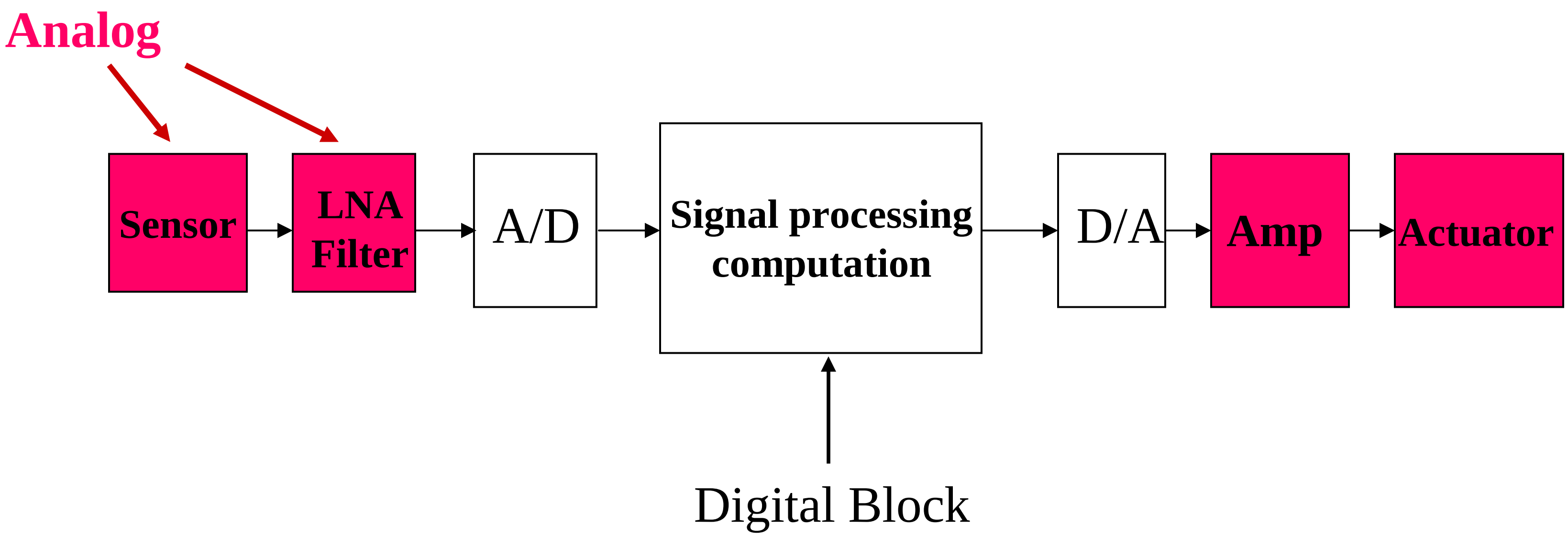
Display
Image

Audio I/O

Analog World

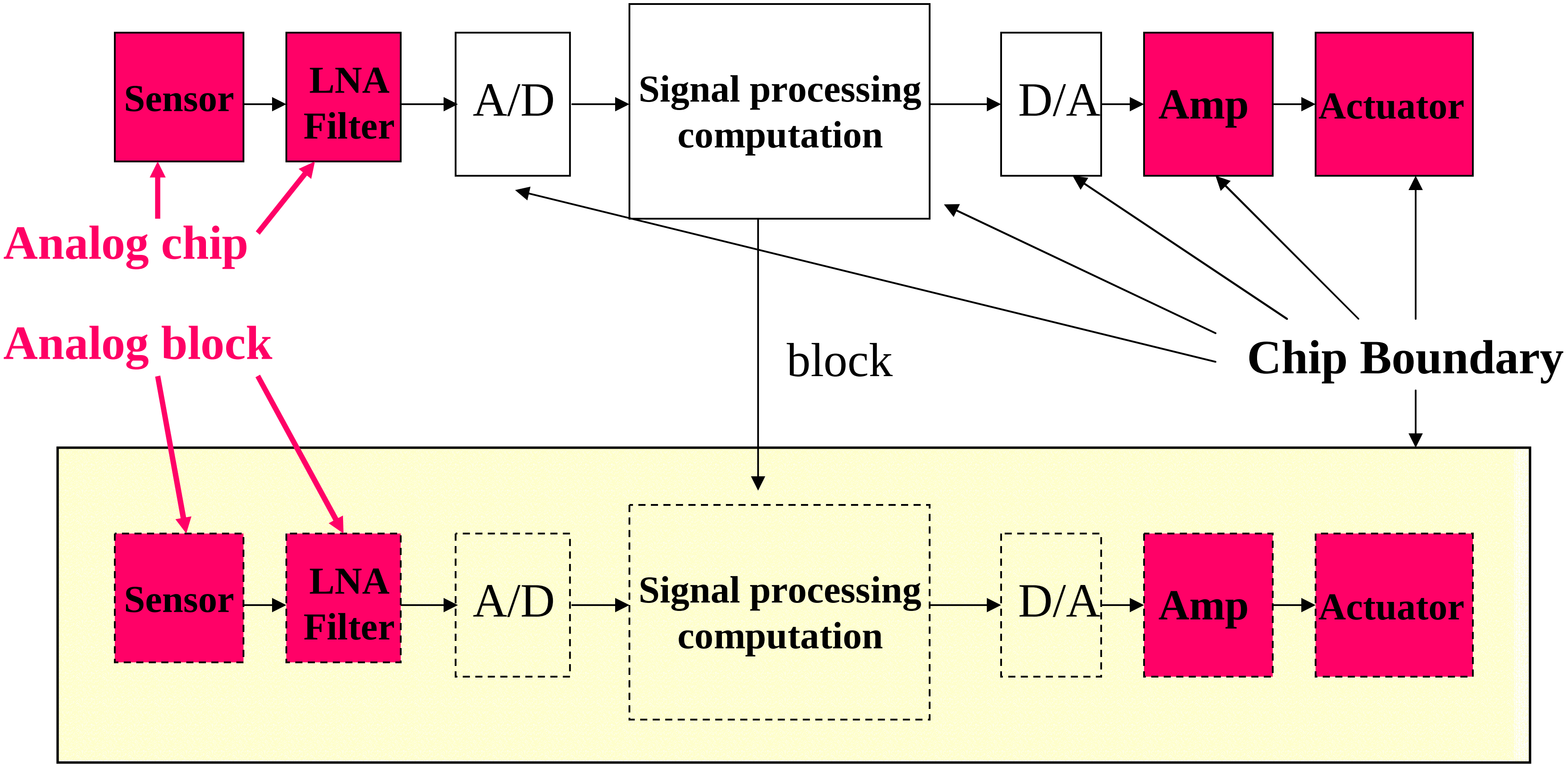
Source Ref : P. R. Gray

Mostly Applied Method of Mixed-Signal Design



系統分割成數顆晶片，每顆晶片分開設計，再經電路板整合。
為確保系統運作之正確，晶片間之界面規格需經嚴謹之定義。

Integration Pushes the Need of Mixed-Signal Design



Benefit of Integration

- Push the limit of system performance

  Reduce parasitic

  Reduce I/O driving loads

  Exploit design space between blocks

- Push the limit of power dissipation

  Reduce parasitic loads

  Reduce I/O driving currents

- Reduce the system size

Challenge of Integration

- High design complexity

  Capacity and Efficiency of EDA tool

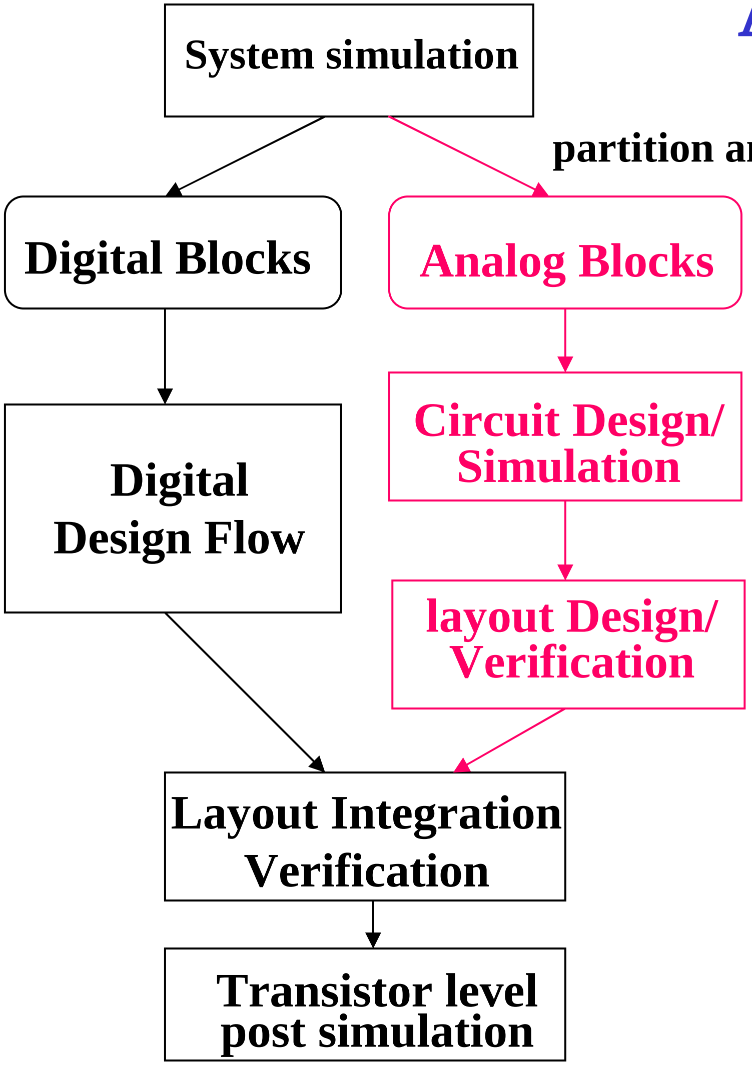
  Different design knowledge
- Increasing process complexity
- Signal coupling prevention

  Signals getting closer

  Signals might be virtually connected
- Signal noise isolation

  Isolation between noisy circuit and sensitive circuit

Conventional Mixed-Signal Design Approach

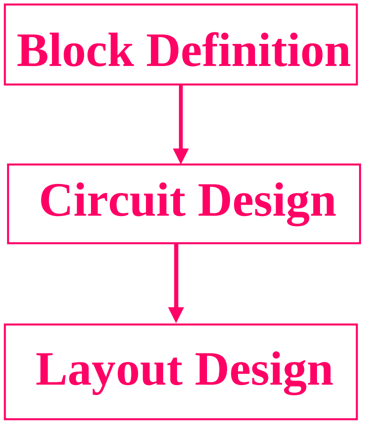


System simulation
partition and spec. definition
Digital Blocks
Analog Blocks
Digital Design Flow
Circuit Design/ Simulation
layout Design/ Verification
Layout Integration Verification
Transistor level post simulation

De-efficiency of the conventional approach

- **The analog / digital design processes are almost independent, lack of horizontal link**
- **The spec. of analog circuit might be over-specified for ensuring correctness of system integration**
- **Hard of analog/MS block reuse evaluation**

Conventional Design Concept for Analog Block



Block Definition

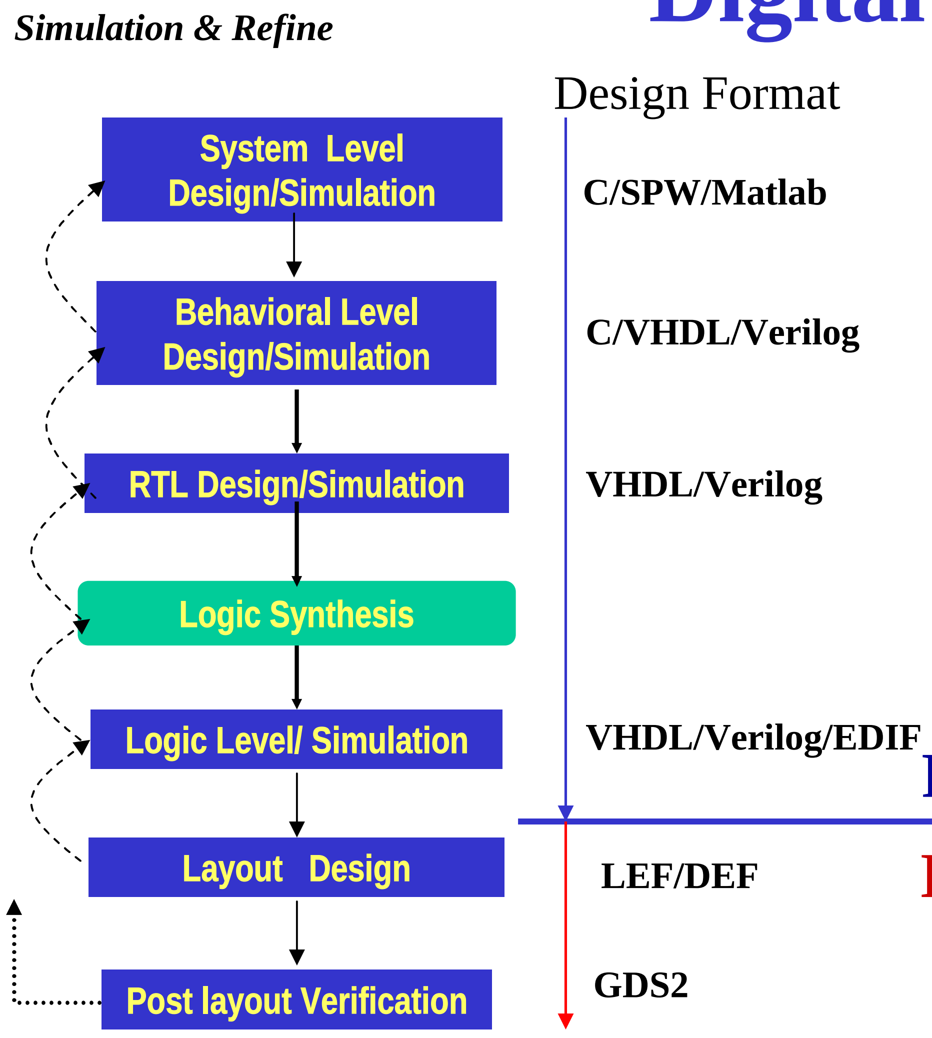
Circuit Design

Layout Design

Problem of Flow :

- Lack of good block description which reflect the complete block characteristics for system simulation
- No efficient translator available for proceeding to the next level
- No good methodology to check the validation of lower level design

Top Down Design Concept in Digital Domain



Simulation & Refine

Design Format

System Level Design/Simulation — C/SPW/Matlab

Behavioral Level Design/Simulation — C/VHDL/Verilog

RTL Design/Simulation — VHDL/Verilog

Logic Synthesis

Logic Level/ Simulation — VHDL/Verilog/EDIF

Layout Design — LEF/DEF

Post layout Verification — GDS2

* At each level, the designed system is simulated to verify the correctness of functionality and performance before proceeding to the next level.

* Tools can be used for the translation of one level to the next level, for example: Behavioral Synthesis, Logic Synthesis, Automatic Place & Route.

* With higher level of abstraction and automation, large system can be designed efficiently.

Front end design

Back end design

Digital Model Abstraction

Digital model maintain the abstraction of system working with discrete events and discrete signal.

System Level

Describe the behavior of entire systems, might include probability analysis.

Behavioral Level

Describe the behavior of blocks of a system, little or no detail on the structure implementation. To prove the basic concepts of the system.

Register Transfer Level

Describe the structure of blocks. Basic components are data storage and operations operate on the stored data.

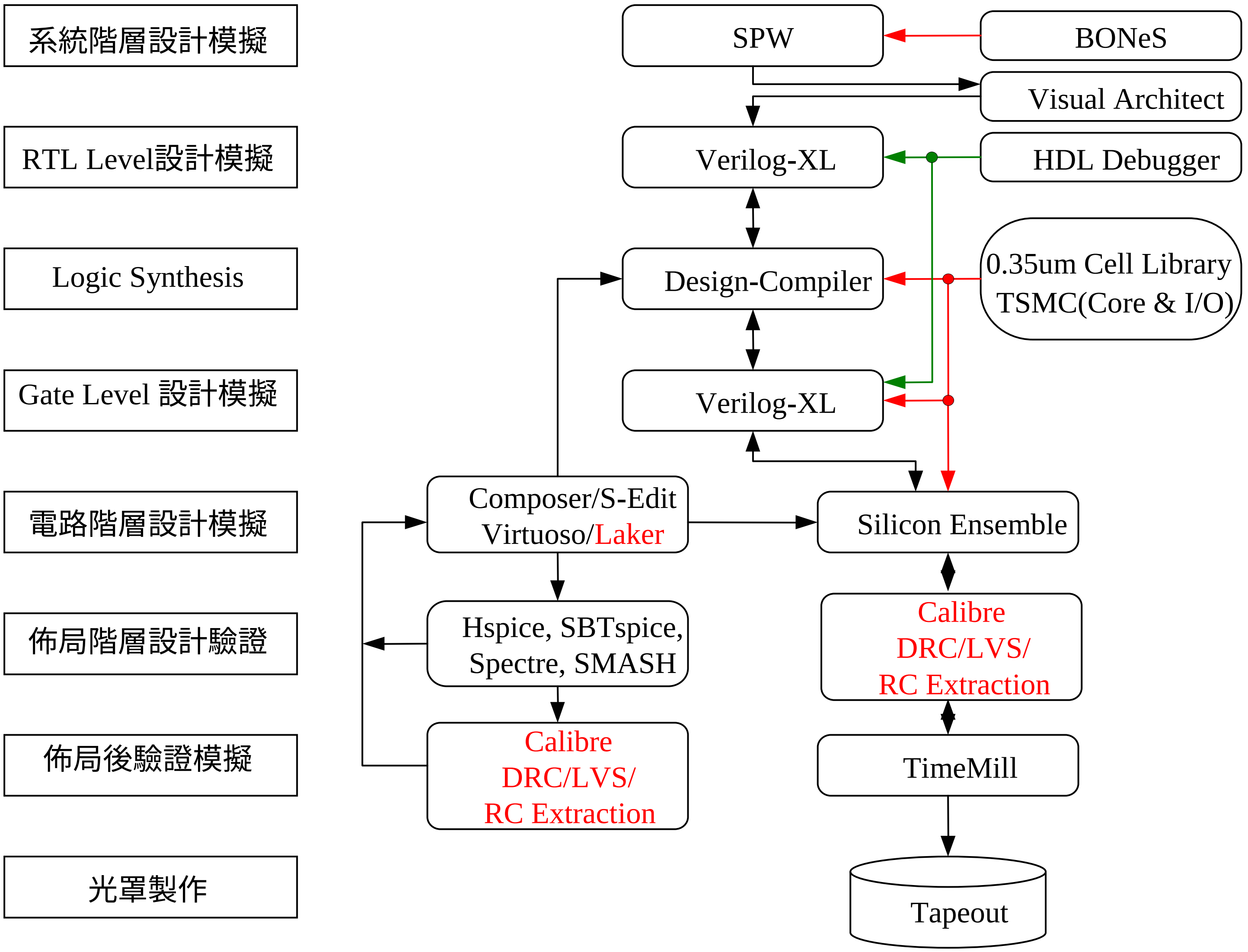
Gate Level

The circuit is described in terms of a set of primitives--Boolean logic with timing data. Timing of individual signal paths can be verified.

Switch Level

The digital logic gates are described in terms of switches -- simplified versions of transistor, detailed timing can be analyzed.

Mixed-Signal Design Flow



系統階層設計模擬

RTL Level設計模擬

Logic Synthesis

Gate Level 設計模擬

電路階層設計模擬

佈局階層設計驗證

佈局後驗證模擬

光罩製作

SPW

BONeS

Visual Architect

Verilog-XL

HDL Debugger

Design-Compiler

0.35um Cell Library
TSMC(Core & I/O)

Verilog-XL

Composer/S-Edit
Virtuoso/Laker

Silicon Ensemble

Hspice, SBTspice,
Spectre, SMASH

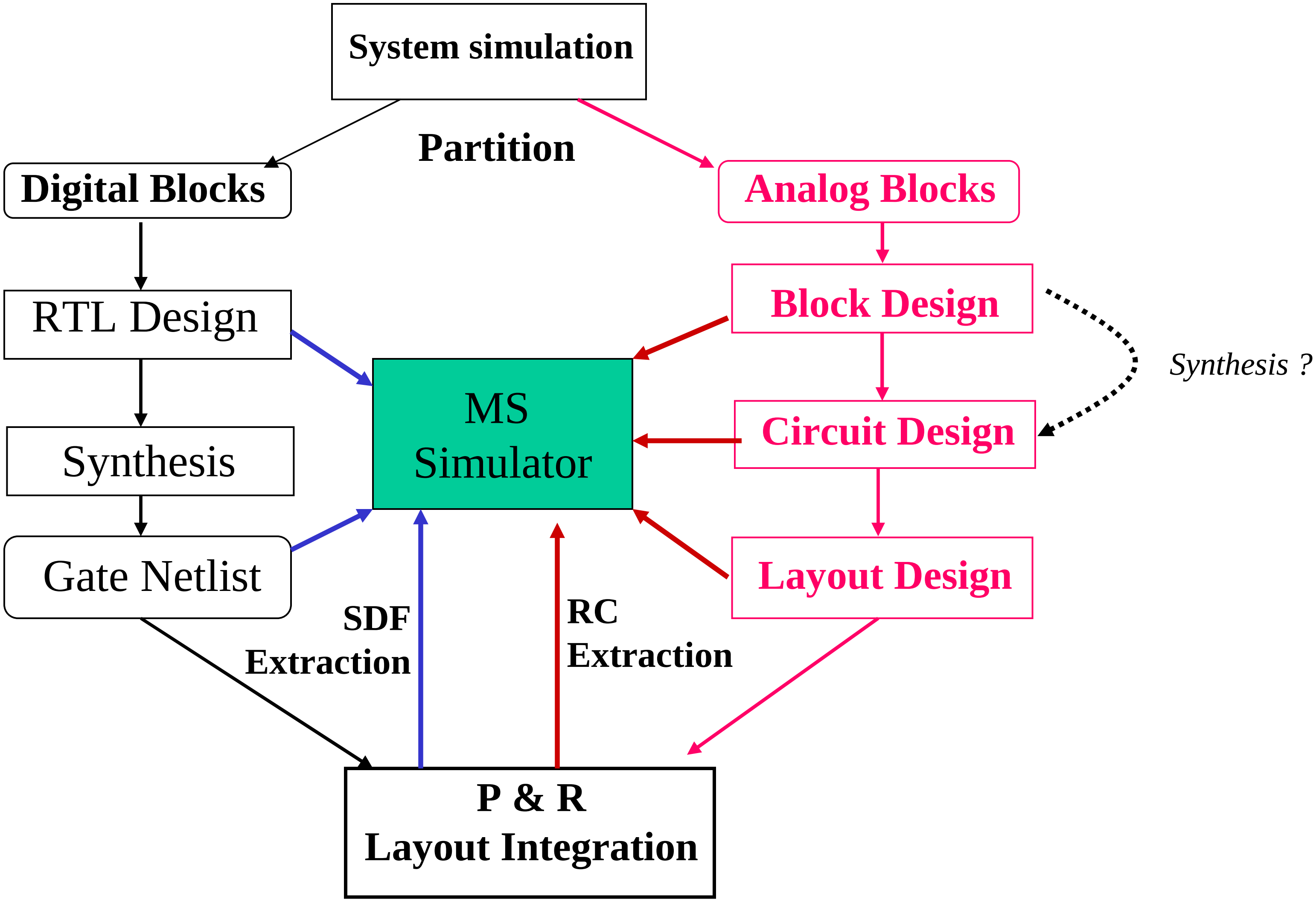
Calibre
DRC/LVS/
RC Extraction

Calibre
DRC/LVS/
RC Extraction

TimeMill

Tapeout

Mixed-Signal Top Down Design Flow

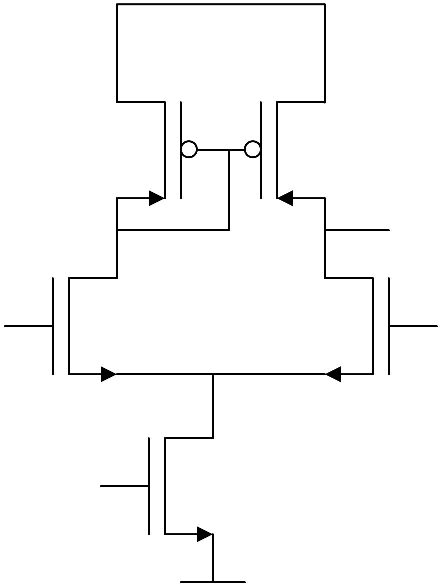


Enabling the Top-Down Design

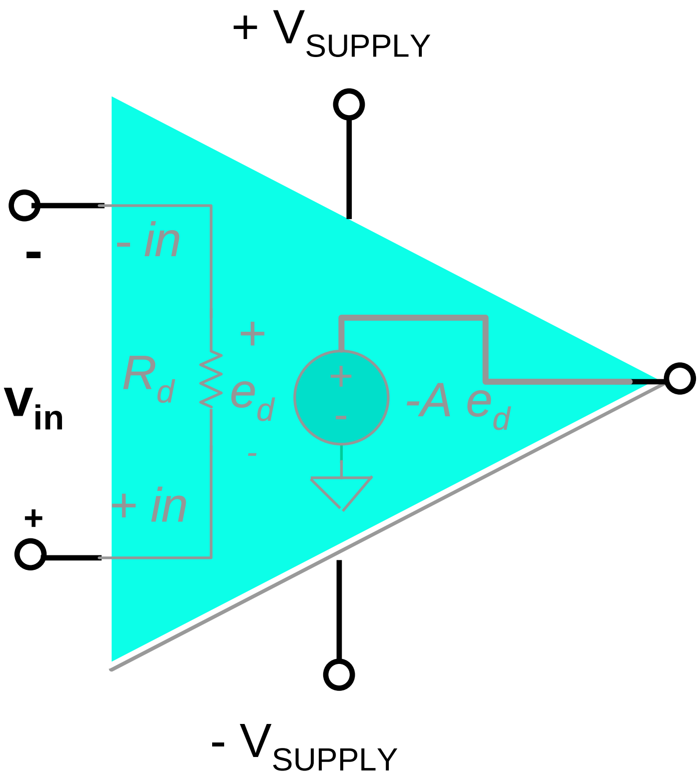
- Behavioral Description Language for Analog/MS Block
- Modeling technique for the Designed Block
- Simulation Capability of handling Behavioral Description
- Simulation Capability of handling Mixed Level simulation
- Simulation Capability of handling Analog/Digital Design

Analog Model Abstraction

Circuit level



Macro Model



Analog Behavioral

```
module opamp (vout, vin_p, vin_n );
  inout  vin_p , vin_n;
  output vout;
  electrical vin_p, vin_n, vout;


  analog begin

     I(vin_p)  <+ 0.0;
     I(vin_n)  <+ 0.0;
     V(vout) <+ V(vin_p, vin_n ) *Av;
  end
endmodule
```

Compromising between Accuracy and Complexity

Top-Down Design Methodology

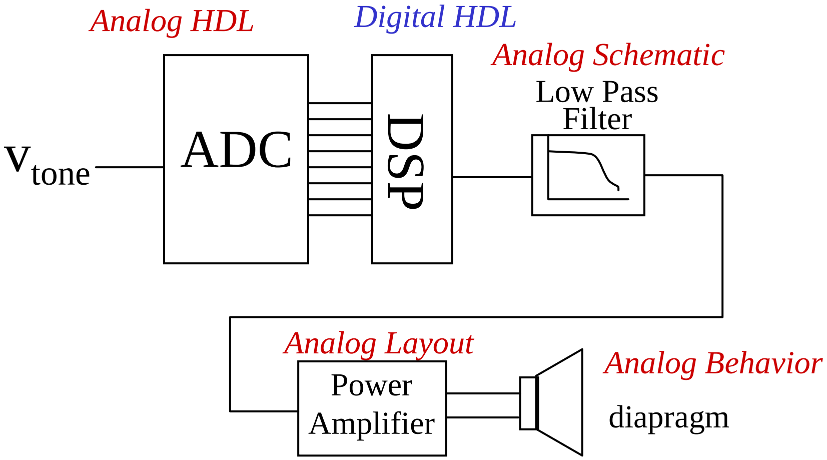
Top-Down Design Methodology

The methodology consists of up-front design and verification of the architecture before creating detailed designs of blocks

Mixed-Level Simulation

One or some blocks at detailed level
Abstract models for remaining blocks

With the aims to :
Improve simulation efficiency
Reduce design iterations



What Can Be Expected with Mixed-Signal Simulation

- Verify the system behavior is correct
- Verify the system requirement is met
- Verify the system performance is satisfied
- Evaluate if certain block architecture is better than others in the system

  The better means :

  - easier to design
  - better performance(area/power/speed/noise)

What Can’t Be Expected with Mixed-Signal Simulation

- Due to the capability of digital simulator, only time domain information can be obtained directly.
- All the modeling of analog behavior should be converted into time domain when simulated with mixed signal simulation.
- Other system characteristics such as frequency response might be calculated from time domain data if needed.
- When obtaining frequency domain information, the time domain information must provide sufficient time period and time point.

What is Required for Mixed-Signal Simulator

Is the model appropriated ?

- Device model supported
- Analog/digital interface

Is the result reliable ?

- Algorithm, methodology

Is the algorithm stable ?

- Ease of convergence

Complete ?

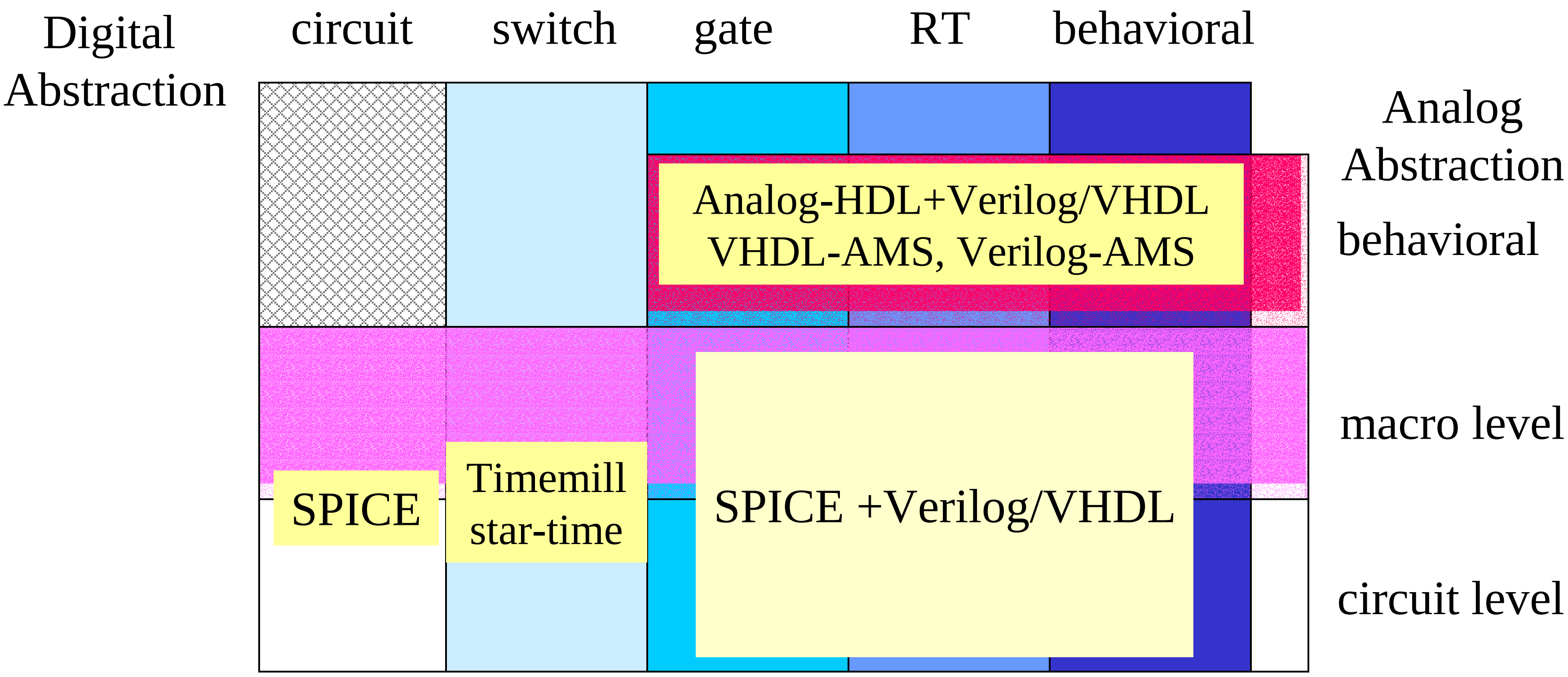
- Design formats, language supported

Is the simulator efficient ?

- Ease of using
- Fast of simulation
- Clear of output result
- Ease of extracting desired parameter

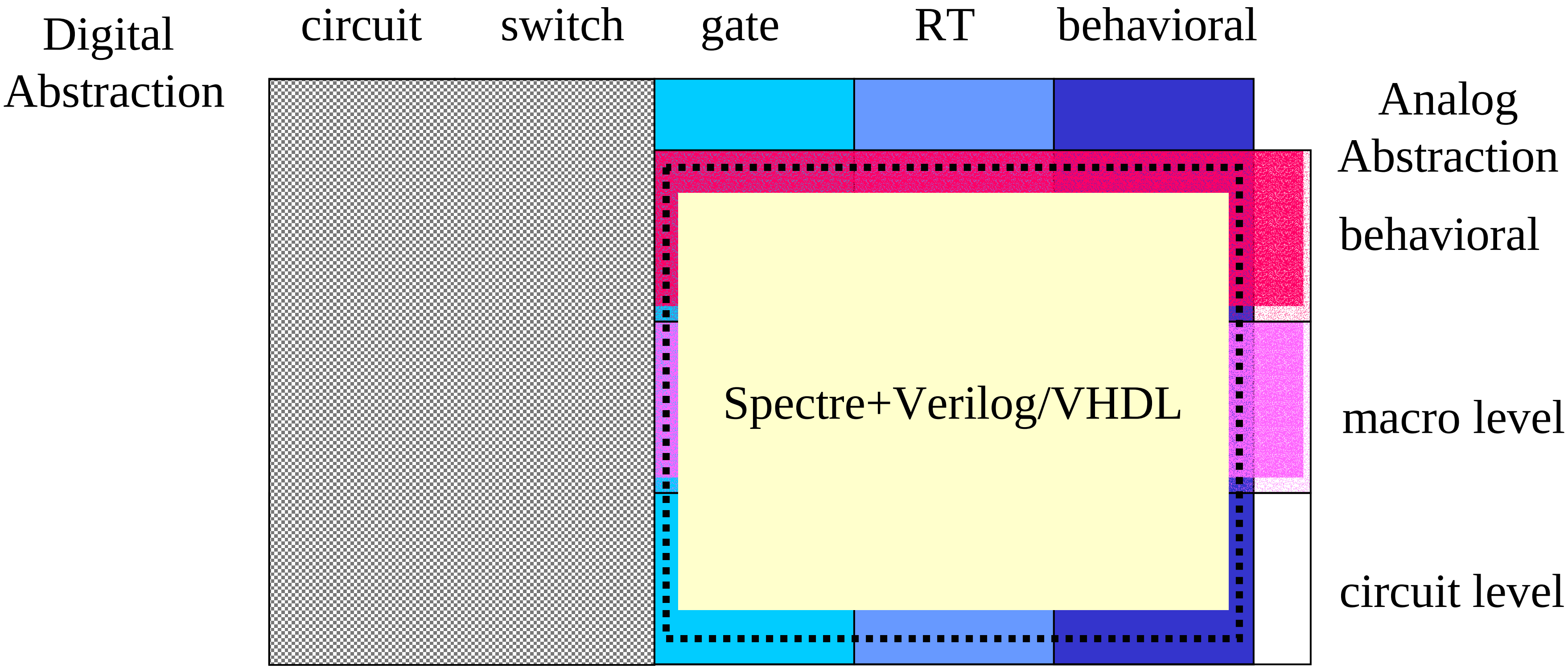
Mixed-Signal Simulation

The most fundamental abstraction for IC design is circuit level.



Multi-Level Mixed-Signal Simulation

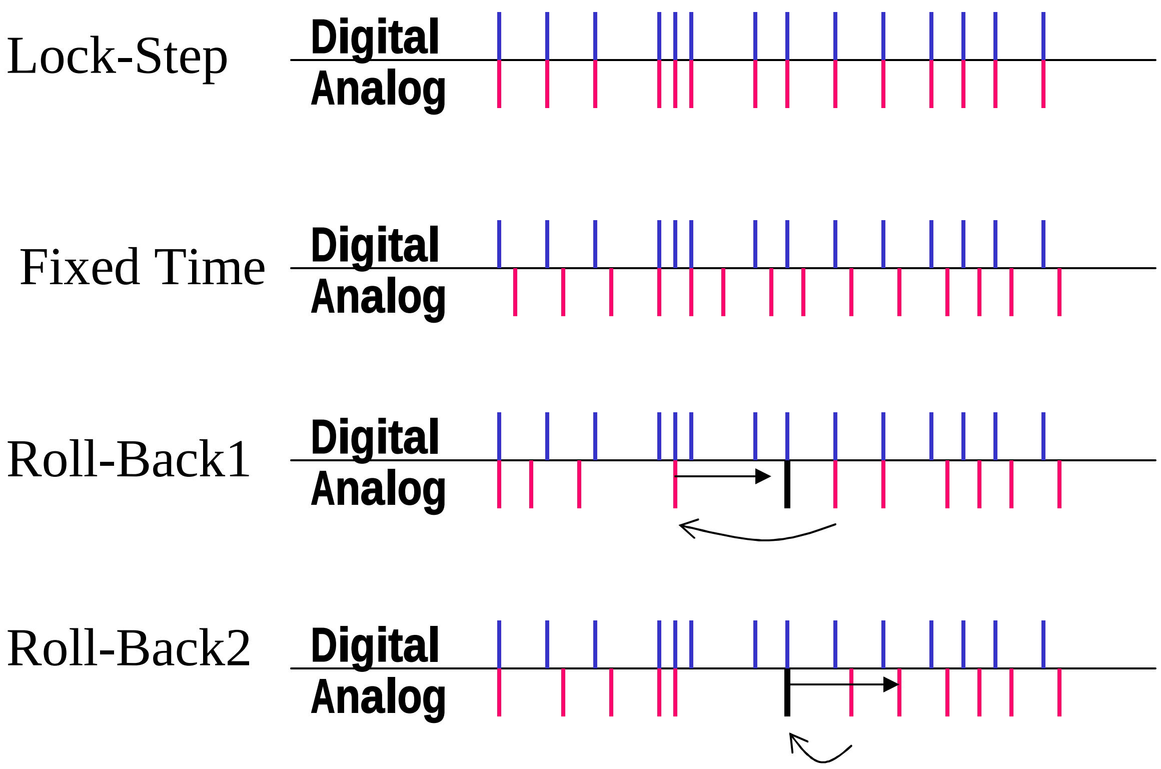
To perform Multi-level Mixed-Signal simulation, simulator must support both circuit level and behavioral level of analog abstraction



Basics of Mixed Signal Simulation

- The fundamentals of Mixed Signal Simulation
  - The identification of analog and digital blocks
    - Signal abstraction : logic value or voltage value
  - The modeling of analog/digital signal translation
    - The loading effect of succeeding block
    - The driving capability of proceeding block
  - The solving of initial solution
  - The types of logic/circuit solver
  - The mechanism of time step control

Timing Control for Mixed-Signal Simulation



Lock-Step

Digital

Analog

Synchronize at minimum time step

Fixed Time

Digital

Analog

Roll-Back1

Digital

Analog

Trace back when a/d, d/a event occurs

Roll-Back2

Digital

Analog

Mixed-Signal Simulator Configurations

Core Modification

Adding extensions to existing simulator for handling mixed-signal design

Glued

Combining two simulator simulating together to achieve simulation
Communication of simulators through Bask Plane or IPC
(Inter- procedure-call)

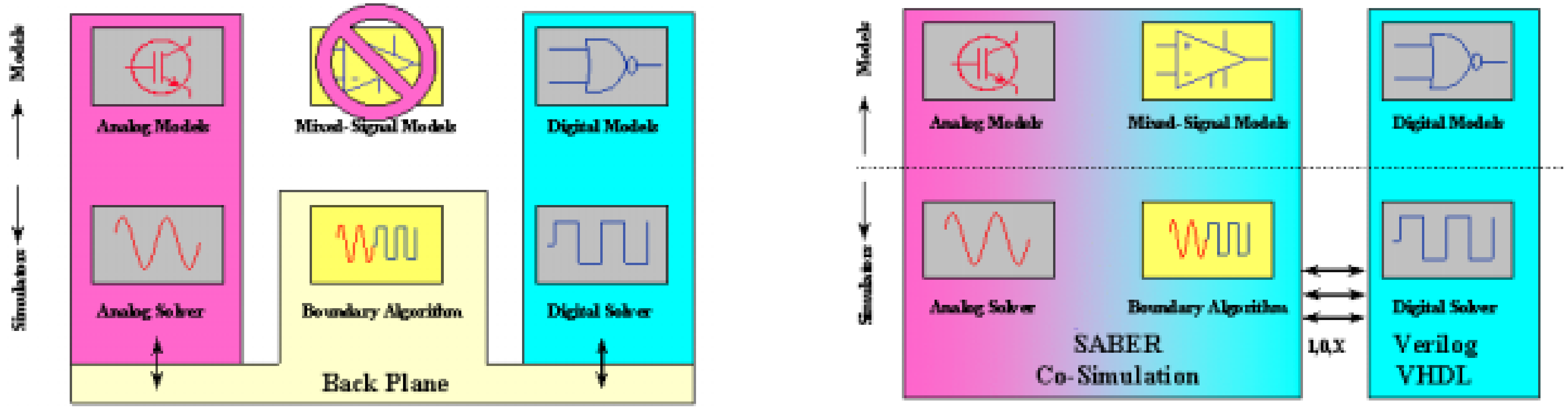


Figure Source: www.vhdl-ams.com

Mixed-Signal Simulator Configurations

Integrated Single Kernel

Use single engine handling different abstractions.
Partitioning and IE handling by system automatically.

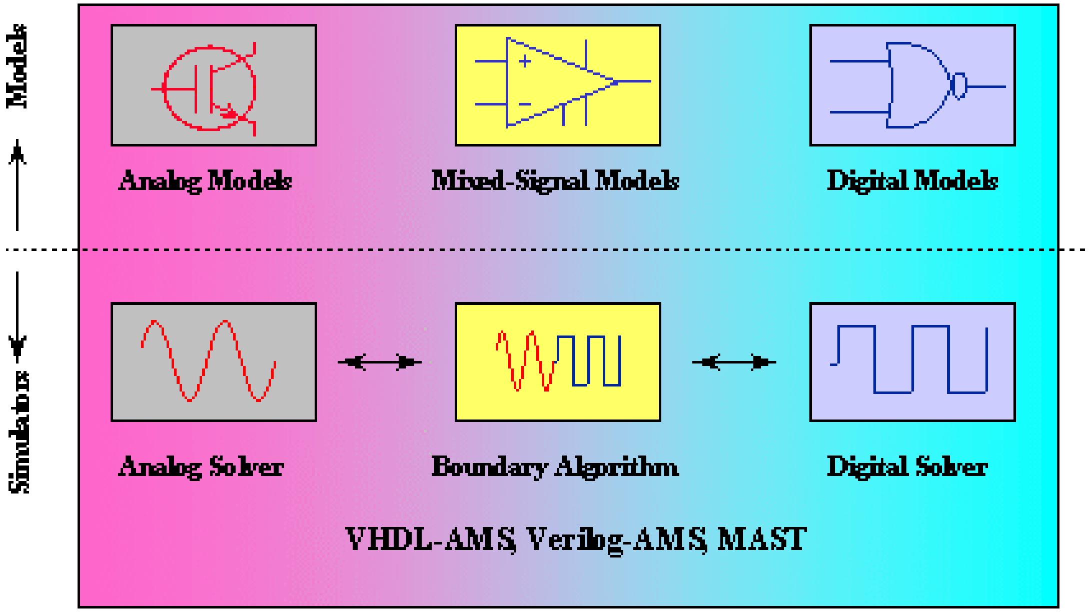


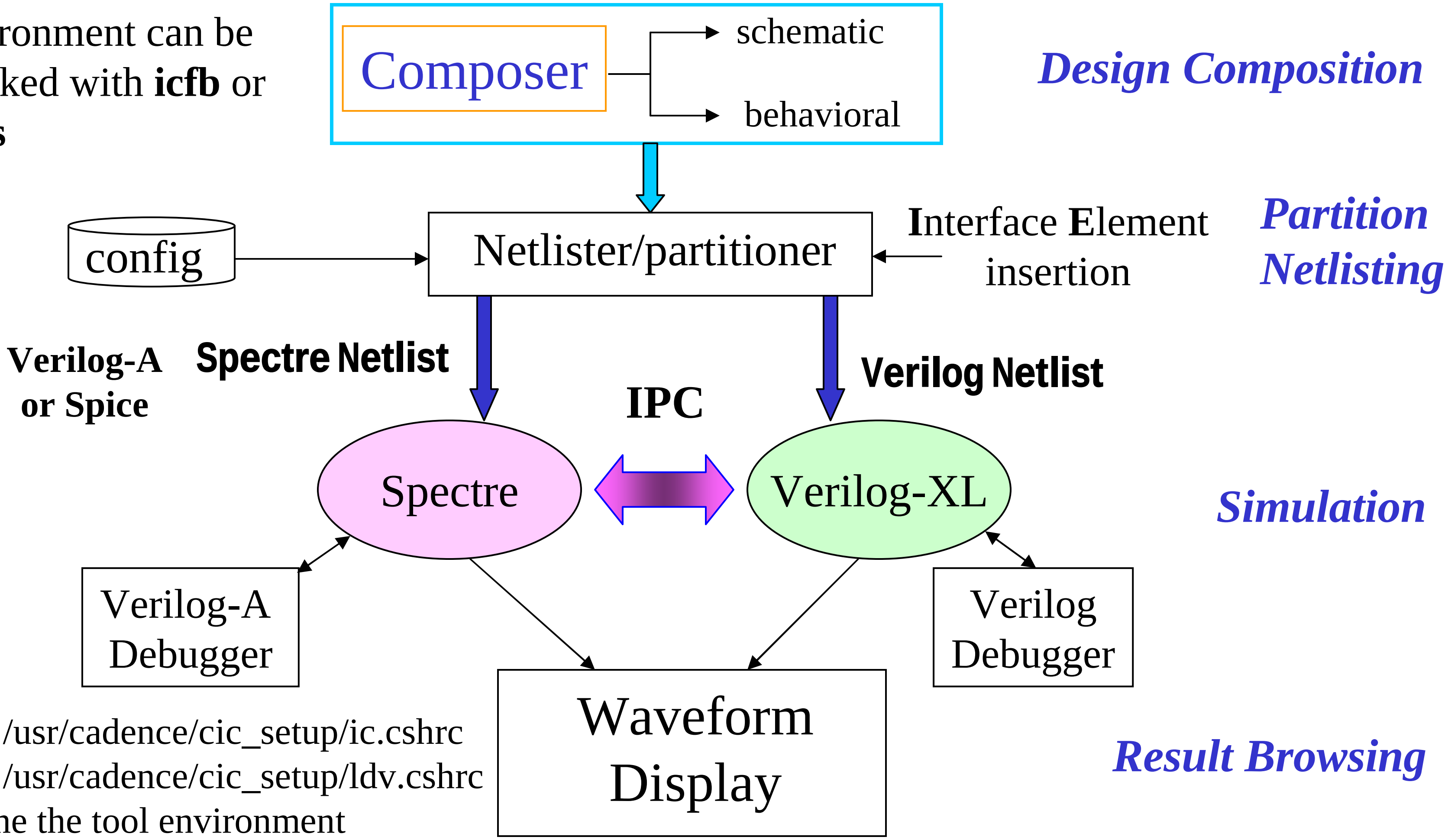
Figure Source: www.vhdl-ams.com

Commercially Available Simulation Environments

- Cadence
  - ***Affirma：VHDL/Verilog, Verilog-A, Spectre***
  - AMS：VHDL/Verilog, Verilog-A, VHDL/Verilog-AMS, Spice/Spectre
- Mentor
  - ADVance MS (ModelSim + Eldo)：C, VHDL/Verilog, Verilog-A, VHDL/Verilog-AMS, Spice
- Synopsys
  - Timemill：Transistor level
  - Star-Sim：Transistor level
  - VCS+NanoSim：C, VHDL/Verilog, Verilog-A, Spice

Affirma Analog Artist Flow

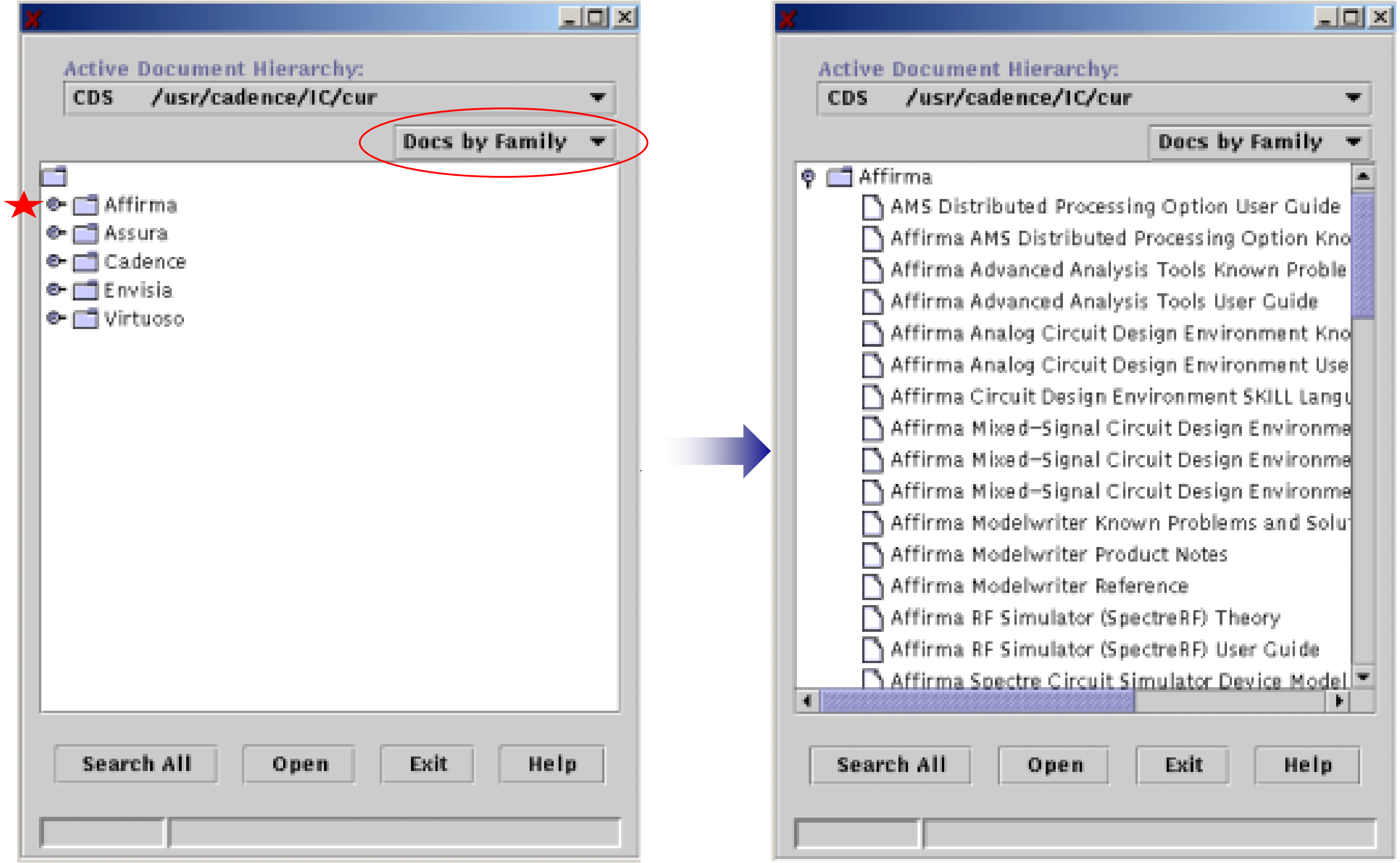
The Cadence environment can be invoked with **icfb** or **icms**



use
source /usr/cadence/cic_setup/ic.cshrc
source /usr/cadence/cic_setup/ldv.cshrc
to define the tool environment

Reference for Affirma Flow

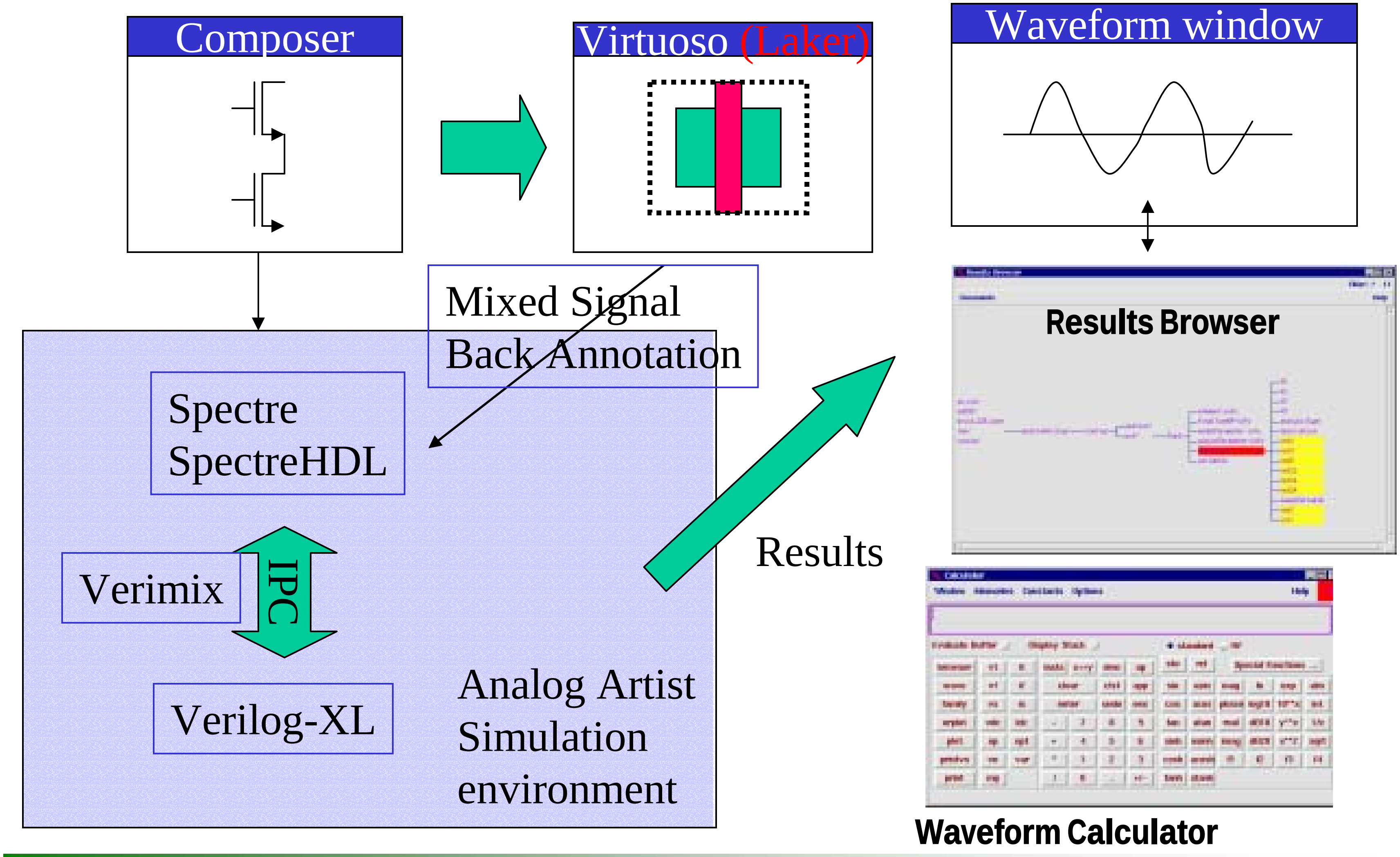
Use **cdsdoc** or help menu from tool window to invoke the manual



Active Document Hierarchy:
CDS /usr/cadence/IC/cur
Docs by Family

- Affirma
  - AMS Distributed Processing Option User Guide
  - Affirma AMS Distributed Processing Option Kno
  - Affirma Advanced Analysis Tools Known Proble
  - Affirma Advanced Analysis Tools User Guide
  - Affirma Analog Circuit Design Environment Kno
  - Affirma Analog Circuit Design Environment Use
  - Affirma Circuit Design Environment SKILL Langu
  - Affirma Mixed-Signal Circuit Design Environme
  - Affirma Mixed-Signal Circuit Design Environme
  - Affirma Mixed-Signal Circuit Design Environme
  - Affirma Modelwriter Known Problems and Solu
  - Affirma Modelwriter Product Notes
  - Affirma Modelwriter Reference
  - Affirma RF Simulator (SpectreRF) Theory
  - Affirma RF Simulator (SpectreRF) User Guide
  - Affirma Spectre Circuit Simulator Device Model

Mixed-Signal Tool Environment



Composer

Virtuoso (Laker)

Waveform window

Mixed Signal Back Annotation

Spectre
SpectreHDL

Verimix

IPC

Verilog-XL

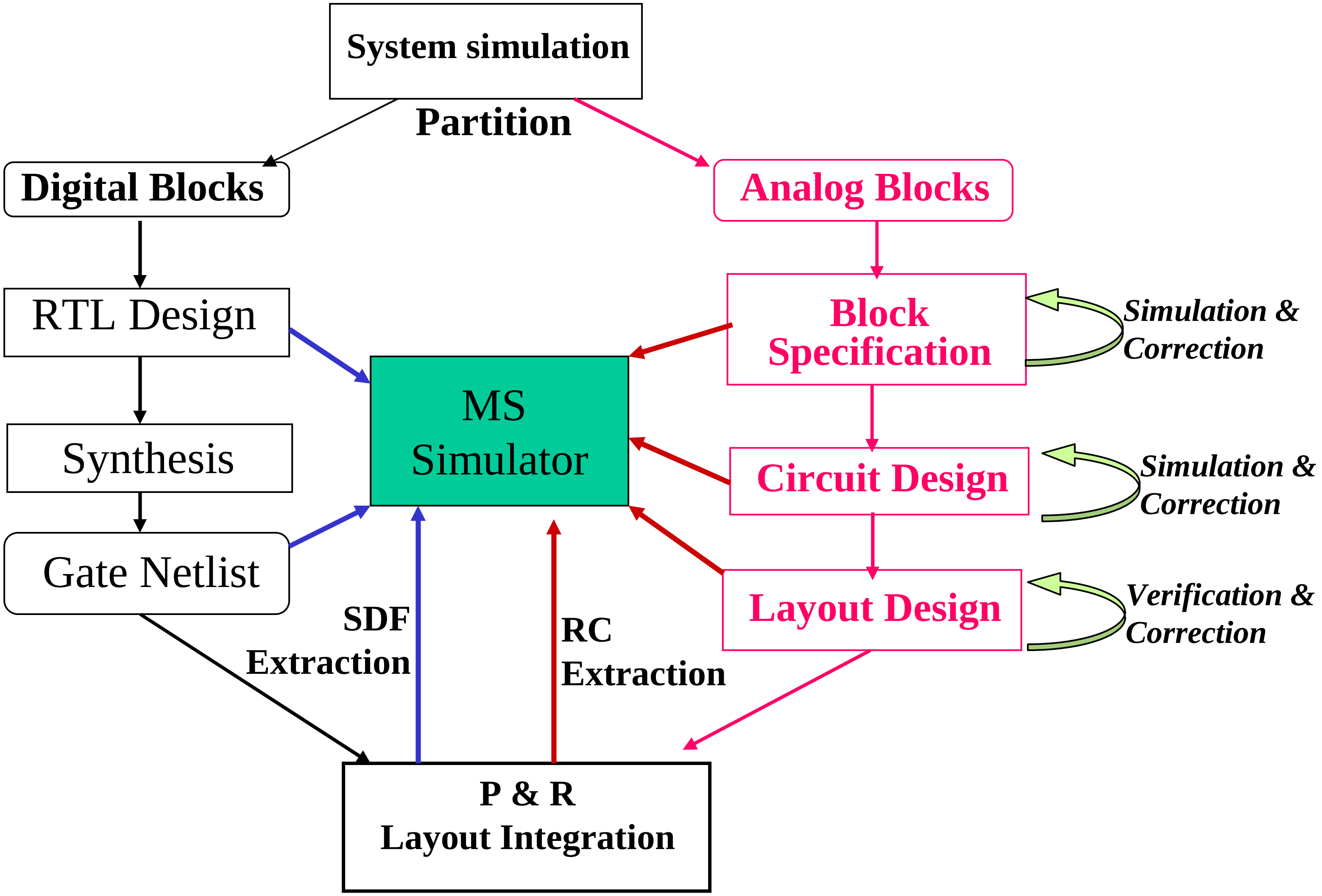
Analog Artist Simulation environment

Results

Results Browser

Waveform Calculator

Mixed-Signal Top Down Design Flow

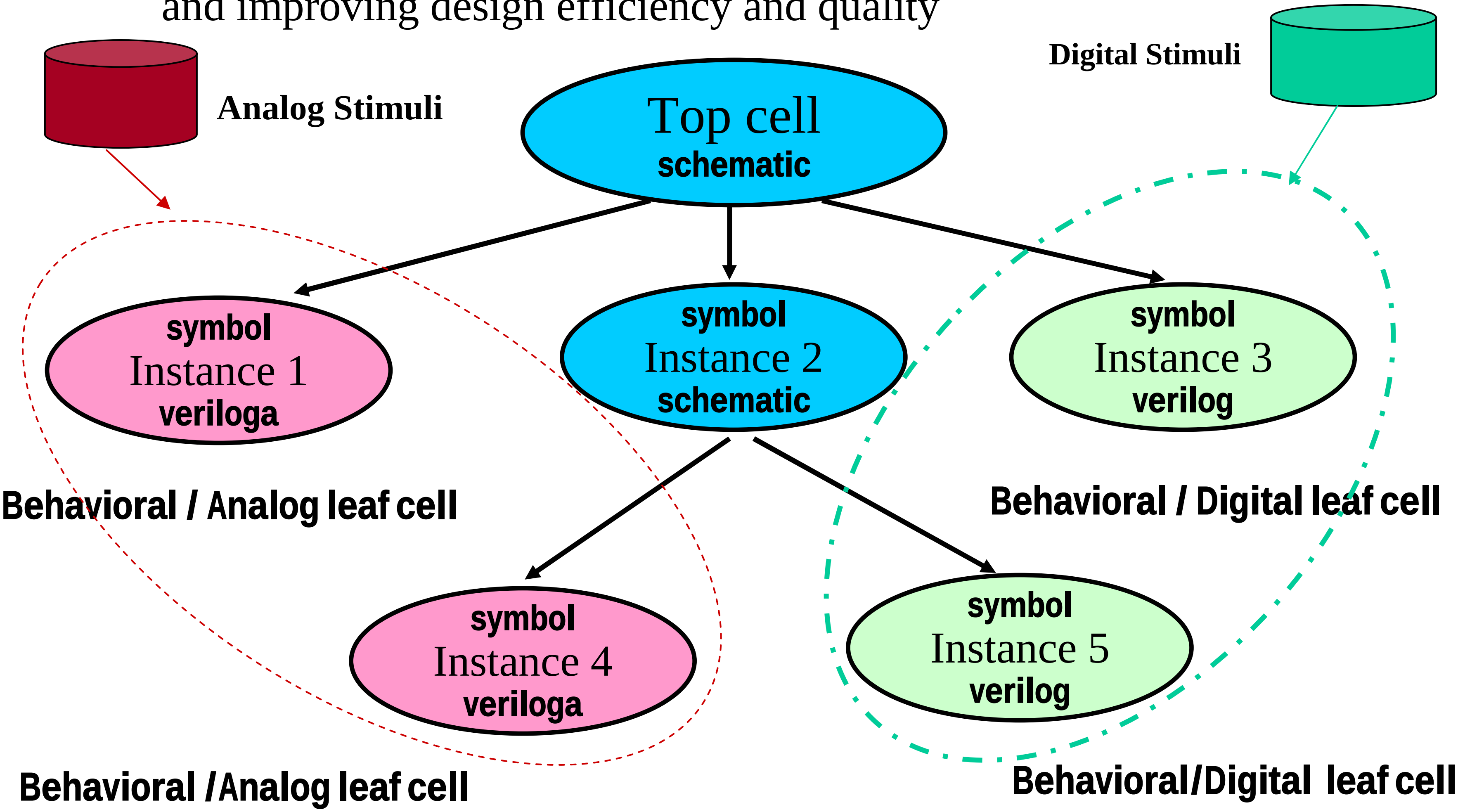


Before the Design Creation

- System Planning and Partitioning
  - Identify digital domain and analog domain, thus the analog/digital interface
  - The better of design partition, the lesser of design iteration and faster of design progression
- Block Boundary definition
  - input/output signals
  - Any input/output characteristics
- Block characteristic definition
  - Algorithm/transfer function
  - constraints/parameters

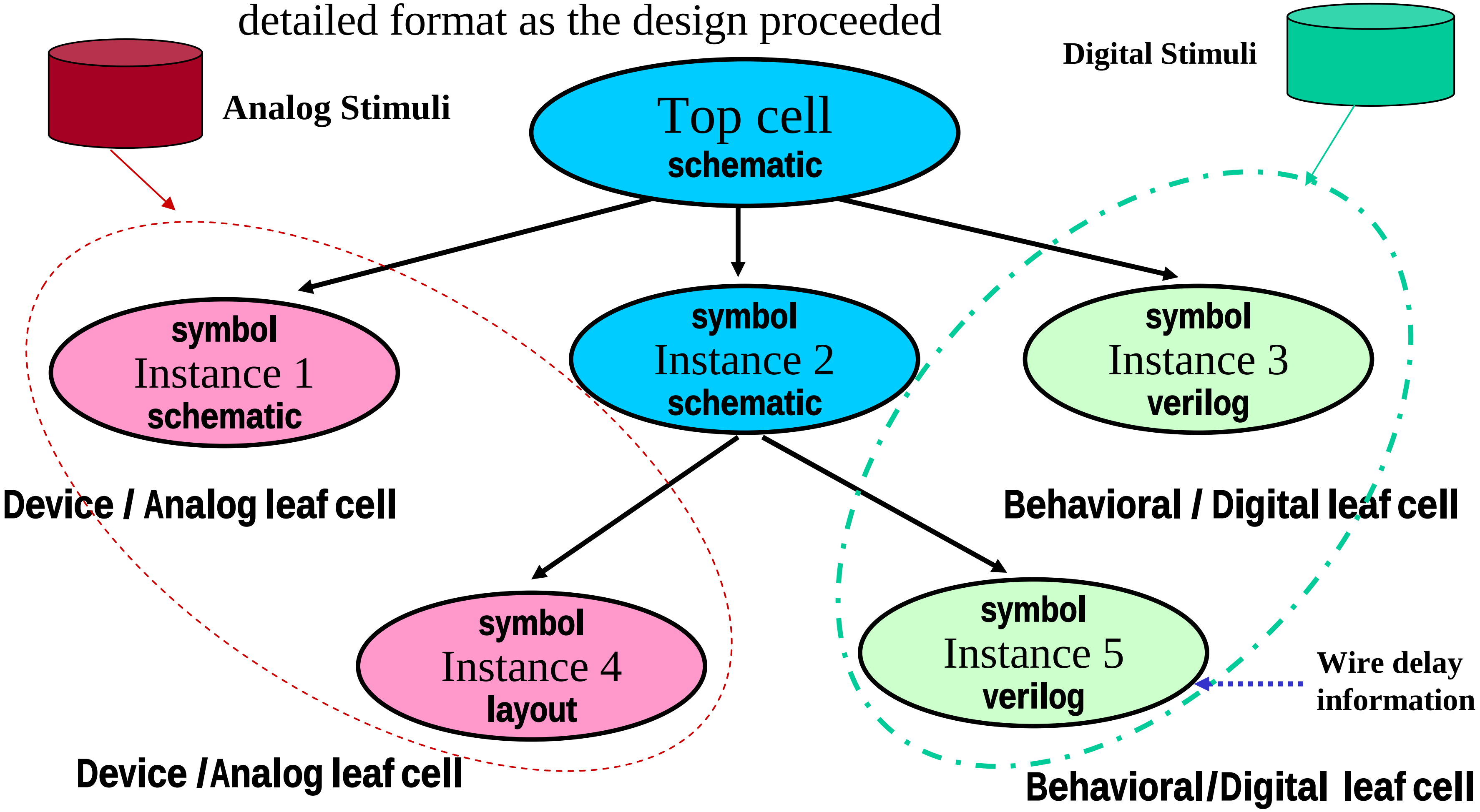
Design Creation

The system is separated into blocks for reducing the design complexity and improving design efficiency and quality



Design Iterations

The block information can be replaced with detailed format as the design proceeded



Design Partitioning Scheme

- The partitioning scheme in Affirma Analog Artist is Instance-based
- Partition is defined before netlisting, each leaf cell must be separated into either digital or analog netlist
- The default partition scheme is based on stop view values

  Analog Stop View Set :

  spectreS cdsSpice spice ahdl auLvs spectre veriloga

  Digital Stop View Set :

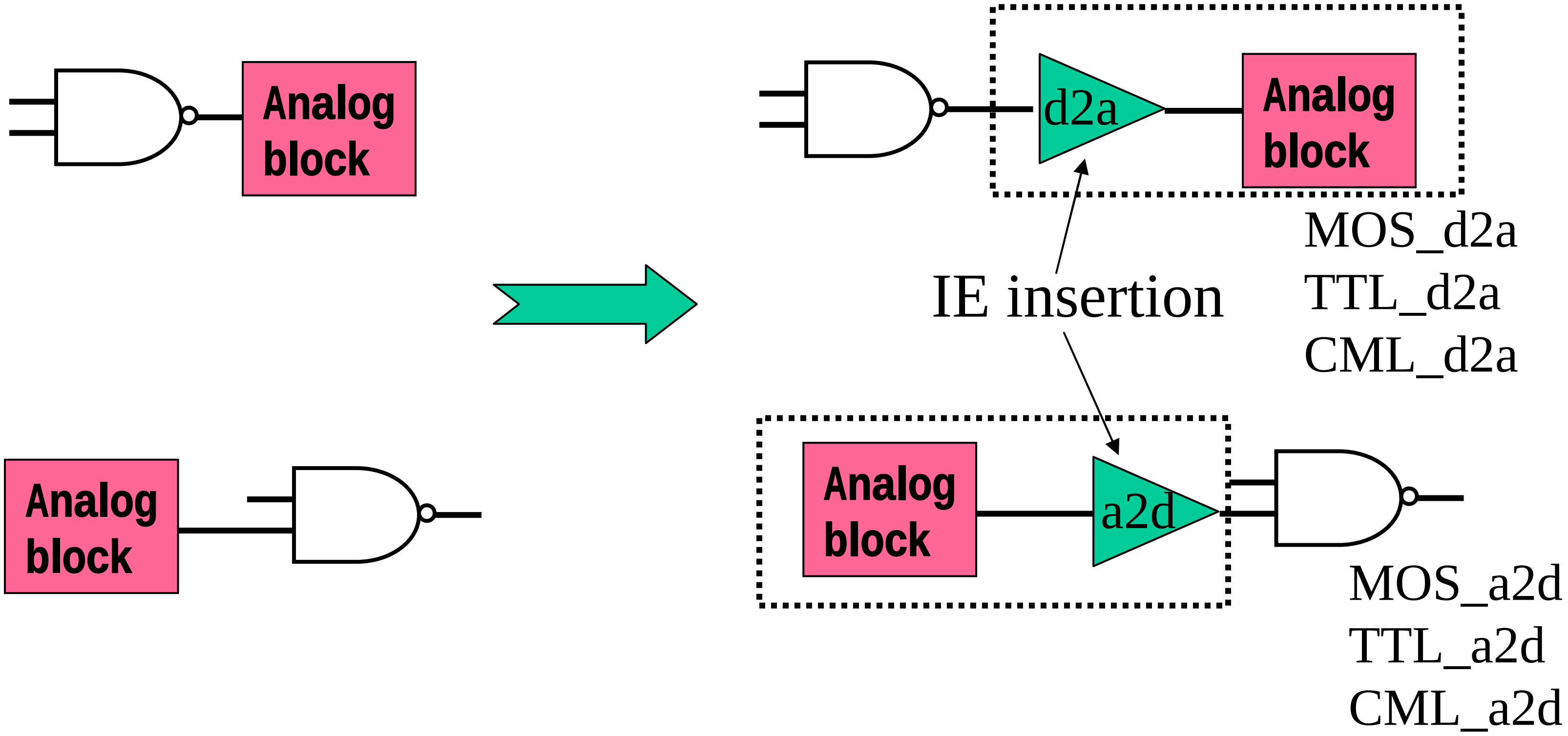
  behavioral functional hdl system verilogNetlist verilog vhdlImport
- The schematic partition can be displayed/defined within schematic window

Partition Requirement

- The design must contain at least one analog component.
- The design must contain at least one digital component.
- There must be with at least one interface net.
- Analog stimuli defined in the analog stimuli file cannot be used to drive digital net.
- Digital stimuli defined in the digital stimuli file can not be used to drive analog net.
- Any interface net must be identified before netlisting.

Interface Element Scheme in Affirma Artist Environment

Interface elements will be inserted during netlist generation for signal translation



Interface Elements

- Generated automatically for input/output terminals of digital components
- Model the loading and driving impedance of digital instance terminals
- Convert voltages to logic levels, and vice versa
- Transport events between two simulators
- No Bi-directional Interface Element provided
- Nonsupply global net can’t be an interface net, ex. clock net can not drive digital and analog blocks simultaneously

Using Analog Artist Environment for Mixed-Signal Design

System verification at the early design stage

Analog Modeling for Mixed Signal Design

- Analog modeling : A key to Top-Down Design methodology

  In order to evaluating performance of system under design, the characteristic of system blocks must be described and included for simulation.

- Macro modeling : Different goal of modeling

  The existed method for analog modeling utilizes the capability of SPICE simulator.
  Analog blocks were described with a set of dependent sources and primitive components.
  Good for modeling a pre-designed circuit with acceptable accuracy
  Bad for block definition at pre-design phase

- HDL modeling : Promoting method for analog/MS design

Analog/Mixed Signal Description Language

- Proprietary Language - MAST, SpectreHDL
- IEEE 1076.1-1999 IEEE VHDL Analog and Mixed Signal Extensions
- OVI Verilog-A 1996
- OVI Verilog Analog/Mixed-Signal (A/MS) 1998

Reference site:

http://www.ovi.org
http://www.eda.org
http://www.vhdl.org

Analog Hardware Description Language Verilog-A

- An extension of the Verilog language to describe analog/mixed signal system models
- To be compatible with Verilog
- An OVI(Open Verilog international) Standard
- An multidiscipline language that models electrical, mechanical, fluid dynamic, and thermodynamic systems
- Can be used for supporting Top-down design

Basic Module Definition

A module represents the fundamental user-defined primitive in Verilog-A

Include natures, discipline & constants

```
`include “constants.h”
`include “discipline.h”
```

Interface Declarations

name, ports and parameters

```
module res1(p, n) ;
inout p, n ;
electrical p, n;
parameter real r=1 from ( 0:inf) ;
parameter real tc=1.5m from [0:3m) ;
```

Global Module Scope

local variables and analog block

Behavioral Description

```
real reff;
analog begin
 @(initial_step(“static”)) begin
   reff = r*(1+tc*$temperature) ;
  end
  I(p,n) <+ V(p, n)/reff ;
end
endmodule
```

Predefined Conservative Disciplines

Defined in disciplines.h

Disciplines	Potential			Flow		
	Nature	Access	Units	Nature	Access	Units
Electrical	Voltage	V	V	Current	I	A
magnetic	**Magnetomotive force**	MMF	A-turn	Flux	Phi	Wb
thermal	**Temperature**	Temp	^{o}C	Power	Pwr	W
kinematics position	Position	Pos	m	Force	F	n
velocity	Velocity	Vel	m/s	Force	F	n
rotational phase	Angle	Theta	rads	Torque	Tau	n/m
velocity	**Angle Velocity**	Omega	rads/s	Torque	Tau	n/m

Verilog-A Modeling Approaches

Structural Model Example

```
module cap(p, n);
inout p, n;
electrical p, n ;
parameter real cvalue = 0 ;
capacitor #(.c(cvalue)) Cmin (p, n) ;
endmodule
```

Behavioral Model Example

```
module cap(p, n);
inout p, n;
electrical p, n ;
parameter real cvalue = 0 ;
analog
   I(p,n) <+ ddt(cvalue*v(p,n)) ;
endmodule
```

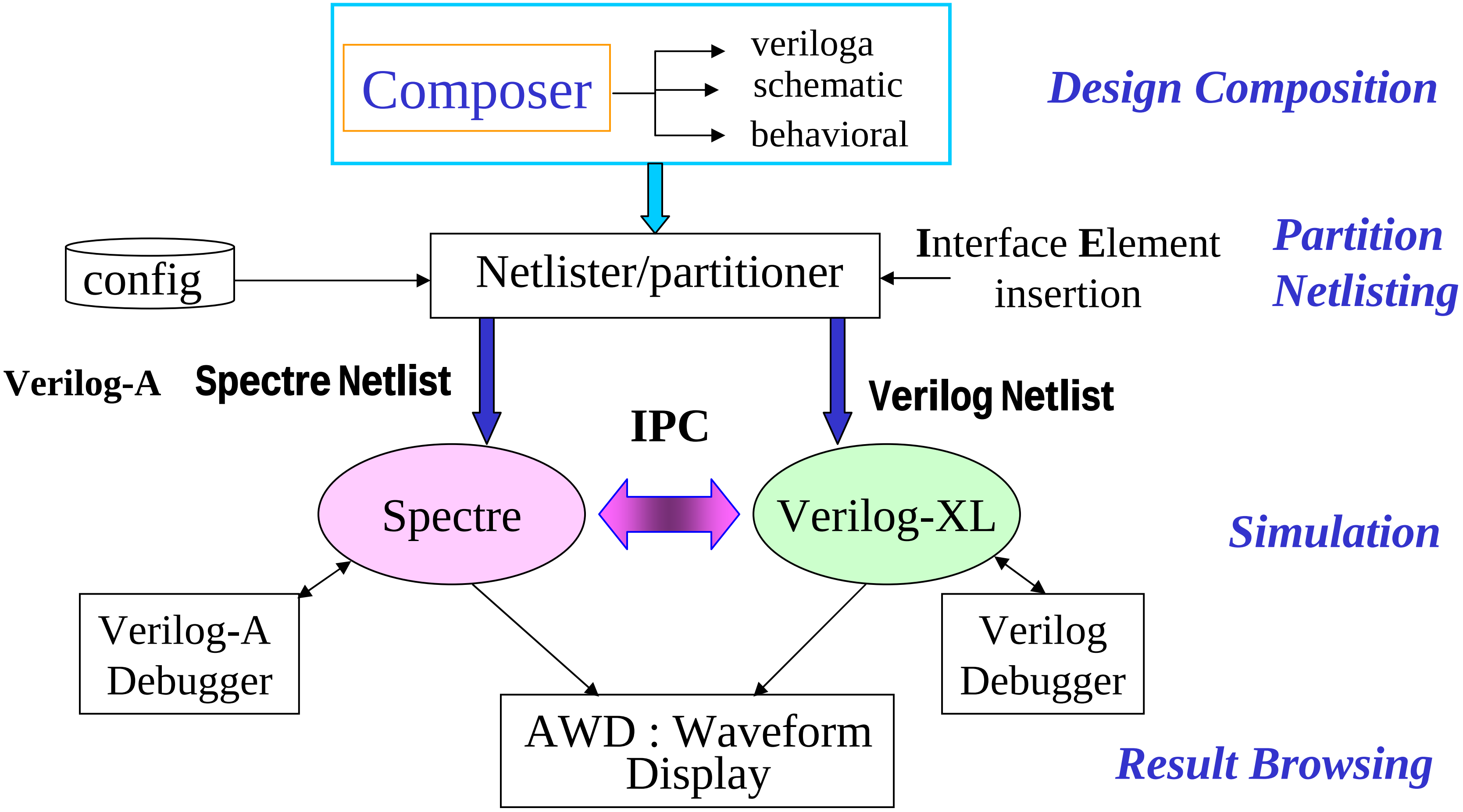
Mixed Structural and Behavioral Models

```
module VCO2(R1, ref, out, CA, CB, VCC, Vcontrol) ;
electrical R1, ref, out, CA, CB, VCC, Vcontrol ;
electrical cntrl ;
 real state ;
 VCOshape shape (ref, cntrl, VCC, Vcontrol) ;
 resistor #(.r(0.001) RX(CB, ref) ;
 resistor #(.r(500) RX(CB, ref) ;
 capacitor #(.c(10p)) Cmin (CA, CB) ;
 analog begin
    @(initial_step) state=1.0 ;
    if ( analysis(“dc”, “static”)) V(CA,CB) <+ 0.0 ;
   @(cross(V(CA)+1.0, -1)) state=1.0 ;
   @(cross(V(CA)-1.0, +1)) state=-1.0 ;
   I(CA) <+ -(1.71*I(cntrl, R1)*V(VCC, ref)*V(out) ;
   V(out) <+ transition(state, 10n, 10n, 10n) ;
 end
endmodule
```

Analog Modeling Issues

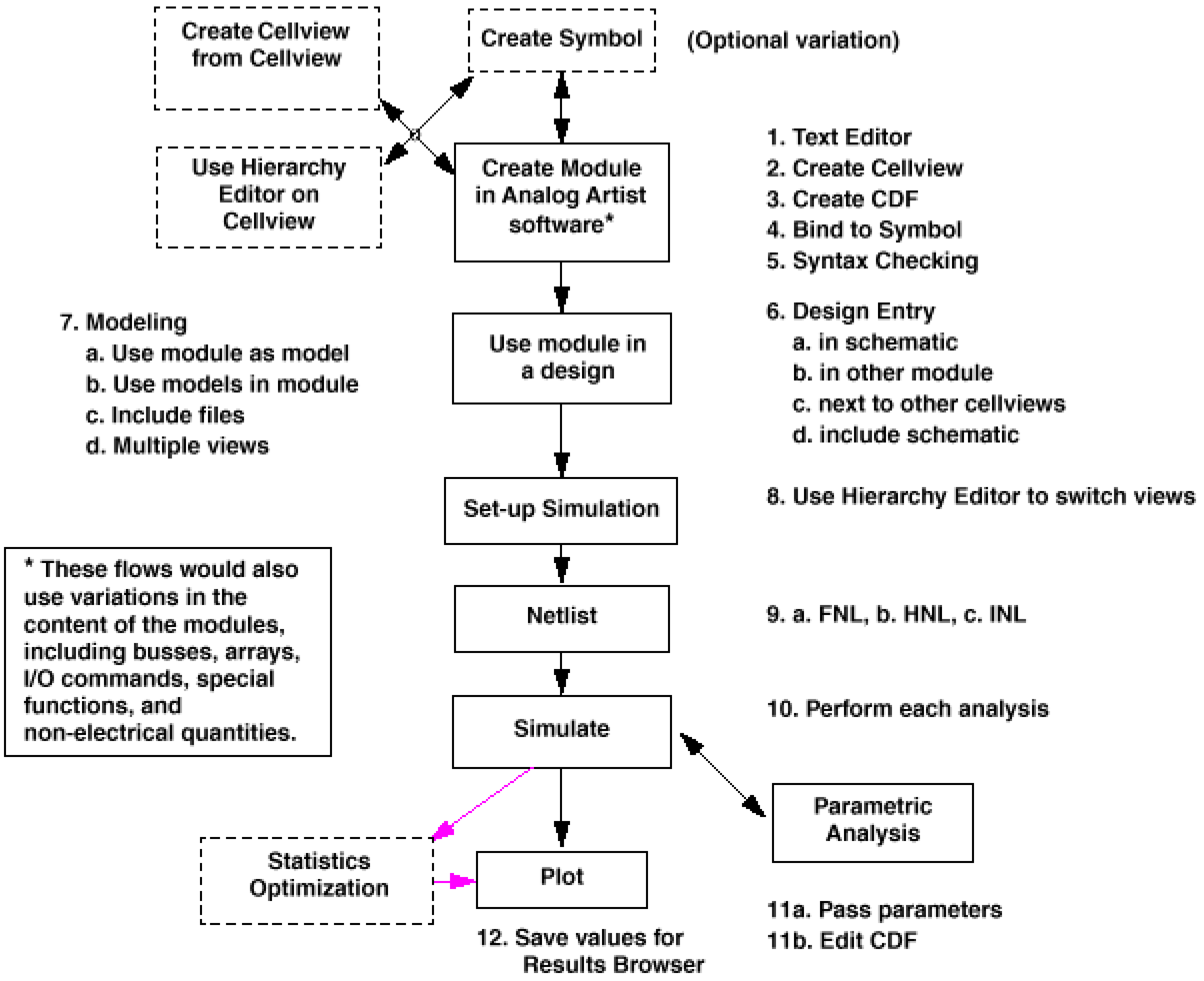
- The Analog/MS Description language provide modeling capability of time domain and frequency domain.
- The analog simulation is solved with Spectre
- Only time domain simulation can be done for mixed signal simulation
- The complete modeling of a block might be difficult, model only as needed.
- Multiple models might be implemented for a single block to describe different view.
- Good model : Simplest form for modeling required information

Mixed Signal Simulation Flow



MS HDL Simulation Flow

Verilog-A in the Analog Artist Design Flow



Create Cellview from Cellview

Create Symbol

(Optional variation)

Use Hierarchy Editor on Cellview

Create Module in Analog Artist software*

Use module in a design

Set-up Simulation

Netlist

Simulate

Parametric Analysis

Statistics Optimization

Plot

1. Text Editor
2. Create Cellview
3. Create CDF
4. Bind to Symbol
5. Syntax Checking

6. Design Entry
   a. in schematic
   b. in other module
   c. next to other cellviews
   d. include schematic

7. Modeling
   a. Use module as model
   b. Use models in module
   c. Include files
   d. Multiple views

8. Use Hierarchy Editor to switch views

9. a. FNL, b. HNL, c. INL

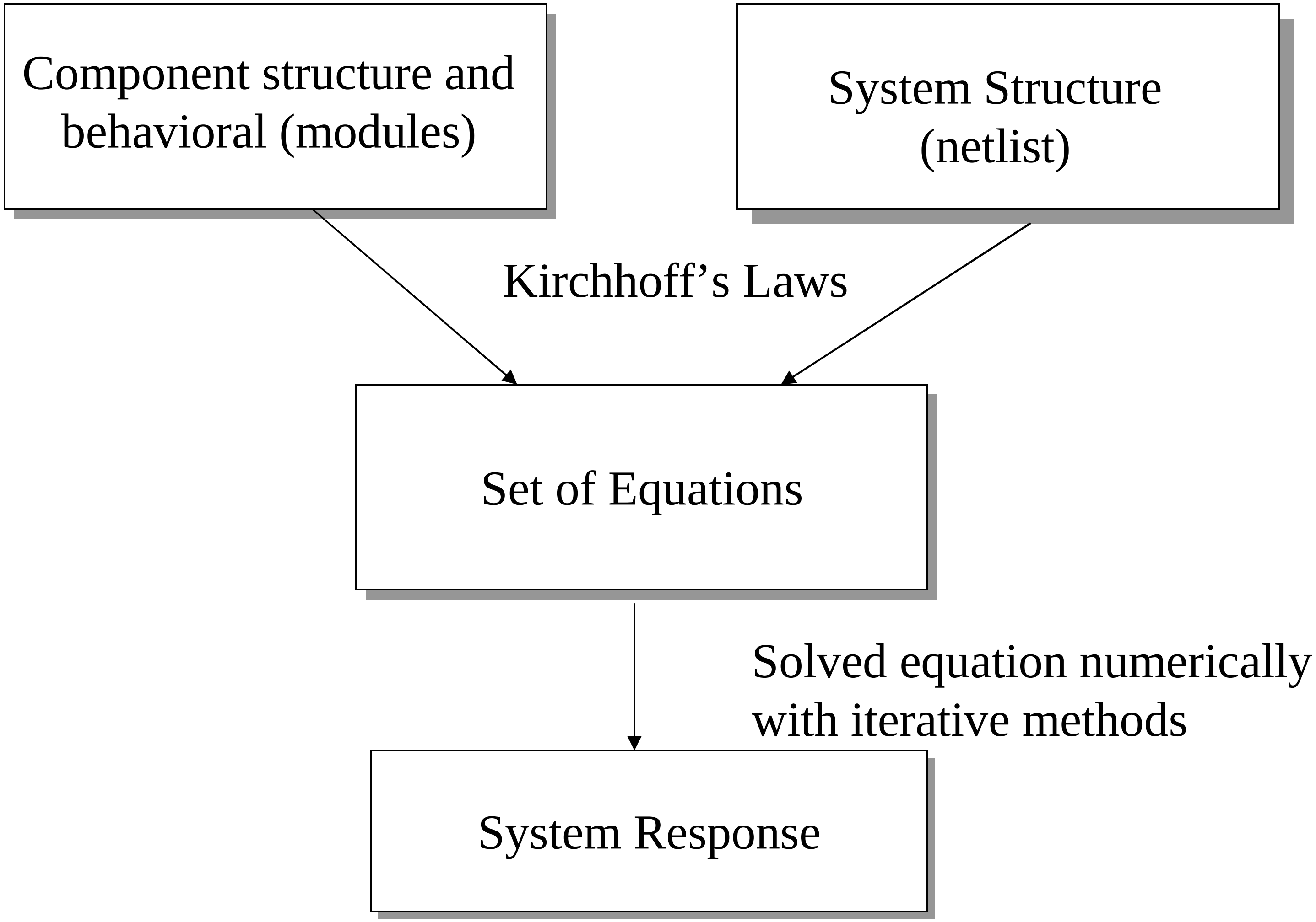
10. Perform each analysis

11a. Pass parameters
11b. Edit CDF

12. Save values for Results Browser

* These flows would also use variations in the content of the modules, including busses, arrays, I/O commands, special functions, and non-electrical quantities.

Describing System



Component structure and behavioral (modules)

System Structure (netlist)

Kirchhoff’s Laws

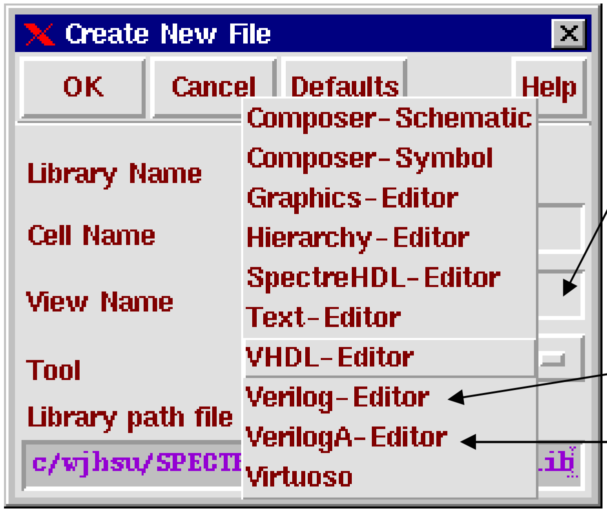
Set of Equations

Solved equation numerically with iterative methods

System Response

Creating the HDL View of Designed Block

- To create a new block, use File → New → Cell View to invoke the create view form



Remember to use **behavioral** as the View name for Verilog

For digital view

For analog view

It is suggested to define the EDITOR variable in your .cshrc file for cadence tool to bring out desired text editor

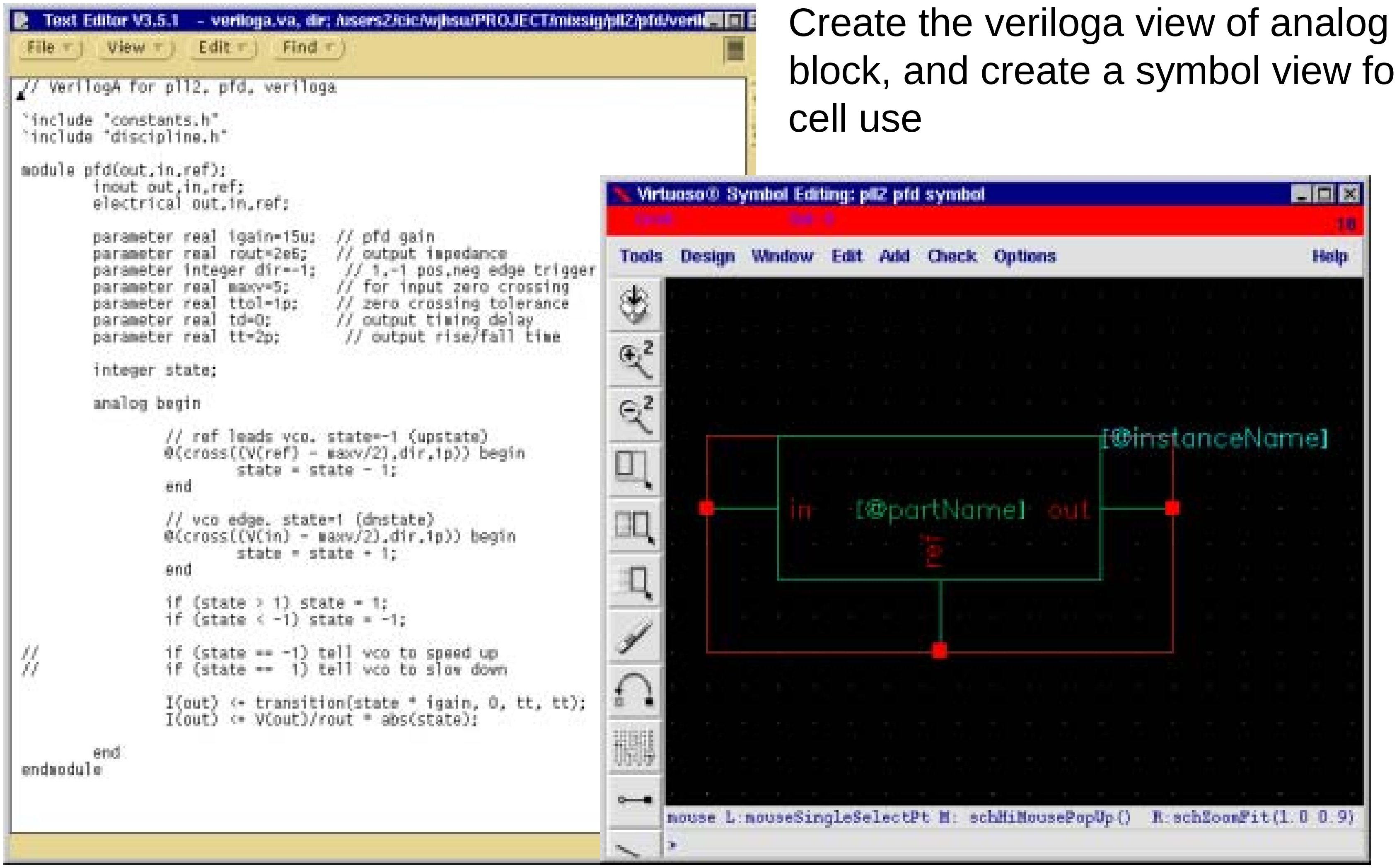
setenv EDITOR textedit

Saving HDL Design

- For behavioral and veriloga view, after saving design and close Textedit window, the system will perform syntax check and extracting port definition.
- The system will not perform syntax check for other view(For example, verilog).
- Error/warning messages will be provided in CIW, if there is any syntax problem in HDL code.
- A symbol view can be generated automatically if the HDL code is correct.

Creating Analog Block and Symbol

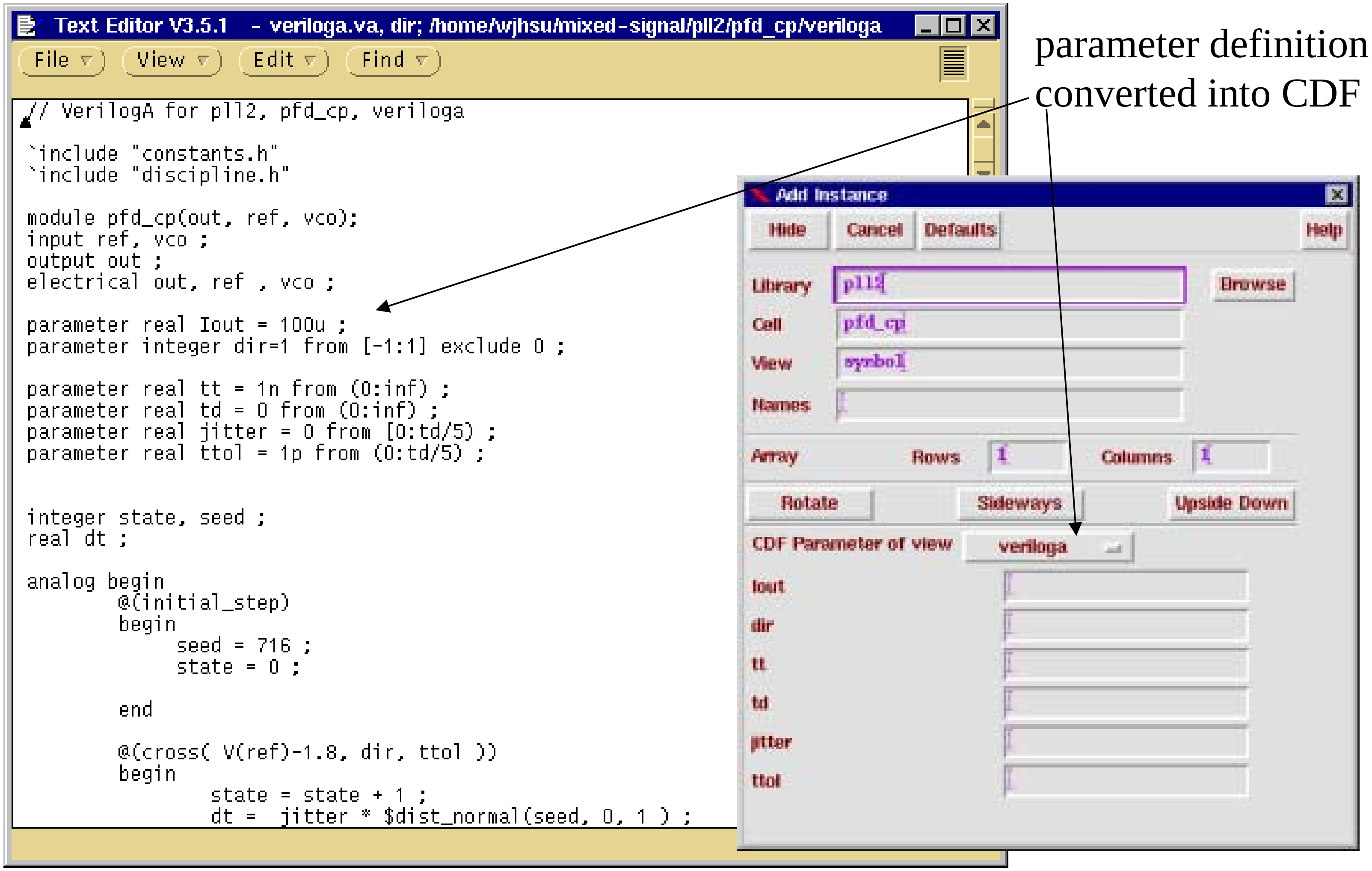
Create the veriloga view of analog block, and create a symbol view for cell use


Text Editor V3.5.1 - veriloga.va, dir: /users2/cic/wjhsu/PROJECT/mixsig/pll2/pfd/veril
File
View
Edit
Find
Virtuoso® Symbol Editing: pll2 pfd symbol
Tools
Design
Window
Edit
Add
Check
Options
Help
[@instanceName]
in
[@partName]
out
ref

```
// VerilogA for pll2, pfd, veriloga

`include "constants.h"
`include "discipline.h"

module pfd(out,in,ref);
        inout out,in,ref;
        electrical out,in,ref;

        parameter real igain=15u;  // pfd gain
        parameter real rout=2e6;   // output impedance
        parameter integer dir=-1;   // 1,-1 pos,neg edge trigger
        parameter real maxv=5;     // for input zero crossing
        parameter real ttol=1p;    // zero crossing tolerance
        parameter real td=0;       // output timing delay
        parameter real tt=2p;       // output rise/fall time

        integer state;

        analog begin

                // ref leads vco. state=-1 (upstate)
                @(cross((V(ref) - maxv/2),dir,1p)) begin
                        state = state - 1;
                end

                // vco edge. state=1 (dnstate)
                @(cross((V(in) - maxv/2),dir,1p)) begin
                        state = state + 1;
                end

                if (state > 1) state = 1;
                if (state < -1) state = -1;

//              if (state == -1) tell vco to speed up
//              if (state ==  1) tell vco to slow down

                I(out) <+ transition(state * igain, 0, tt, tt);
                I(out) <+ V(out)/rout * abs(state);

        end
endmodule
```

```
mouse L:mouseSingleSelectPt M: schHiMousePopUp()  R:schZoomFit(1.0 0.9)
>
```

Parameter Definition


Text Editor V3.5.1 - veriloga.va, dir; /home/wjhsu/mixed-signal/pll2/pfd_cp/veriloga
File
View
Edit
Find
Add Instance
Hide
Cancel
Defaults
Help
Browse
Rotate
Sideways
Upside Down

```
// VerilogA for pll2, pfd_cp, veriloga

`include "constants.h"
`include "discipline.h"

module pfd_cp(out, ref, vco);
input ref, vco ;
output out ;
electrical out, ref , vco ;

parameter real Iout = 100u ;
parameter integer dir=1 from [-1:1] exclude 0 ;

parameter real tt = 1n from (0:inf) ;
parameter real td = 0 from (0:inf) ;
parameter real jitter = 0 from [0:td/5) ;
parameter real ttol = 1p from (0:td/5) ;


integer state, seed ;
real dt ;

analog begin
        @(initial_step)
        begin
             seed = 716 ;
             state = 0 ;

        end

        @(cross( V(ref)-1.8, dir, ttol ))
        begin
               state = state + 1 ;
               dt =  jitter * $dist_normal(seed, 0, 1 ) ;
```

Library: p112
Cell: pfd_cp
View: symbol
Names:
Array Rows: 1 Columns: 1
CDF Parameter of view: veriloga
Iout:
dir:
tt:
td:
jitter:
ttol:

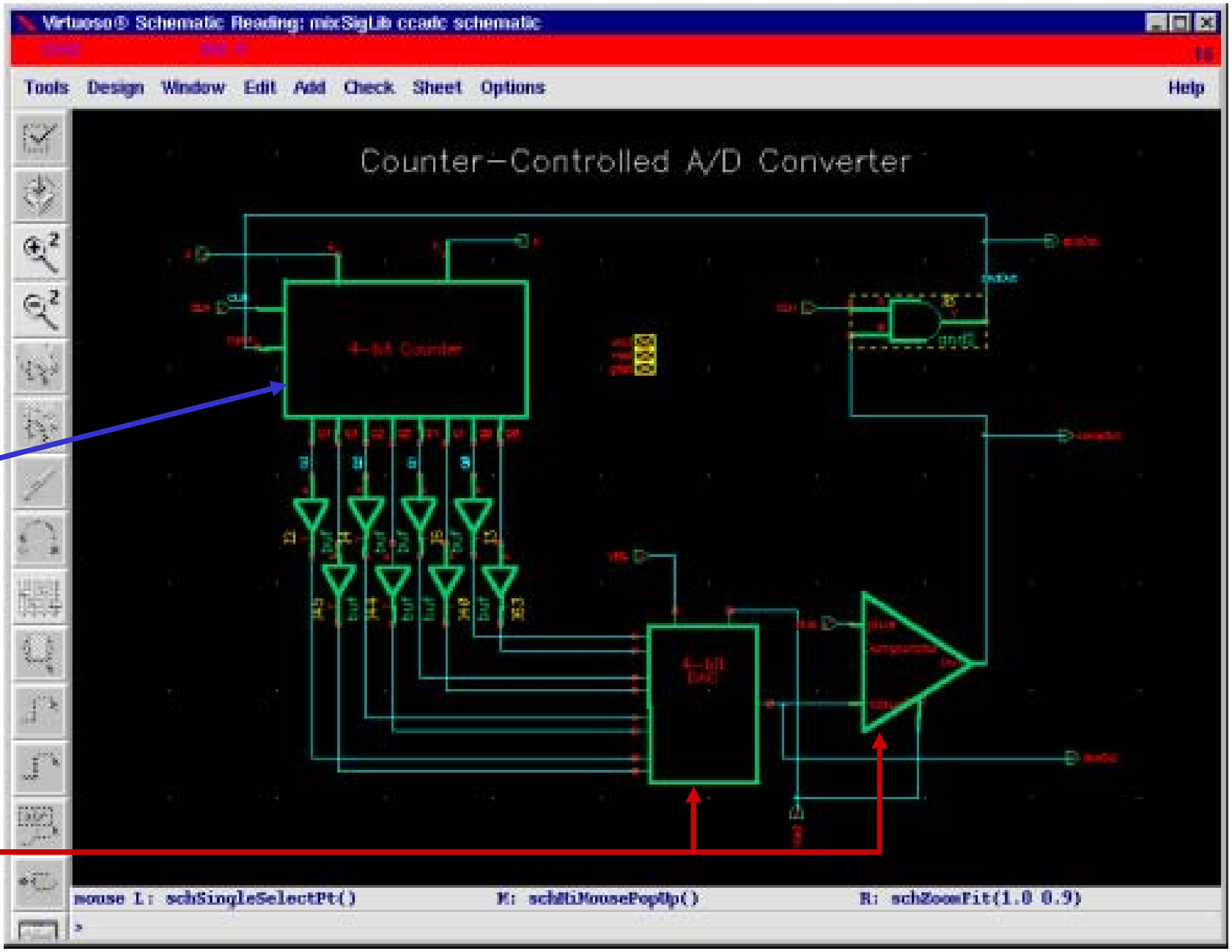
parameter definition will be converted into CDF parameter

Integrating Design Using Composer

Integrating the whole design within a schematic window, this will be top view of design

digital block

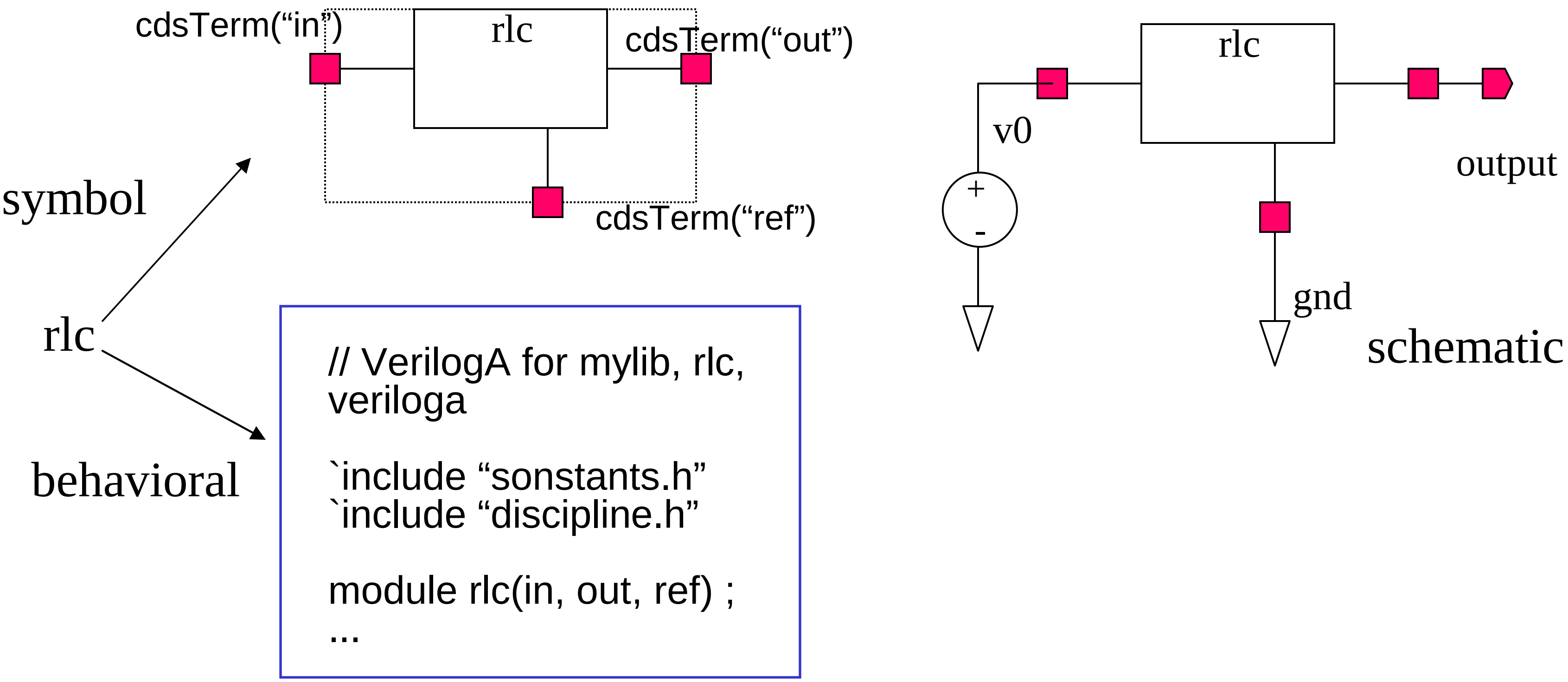
analog block


Virtuoso® Schematic Reading: mixSigLib ccadc schematic
Tools
Design
Window
Edit
Add
Check
Sheet
Options
Help
Counter–Controlled A/D Converter
4-bit Counter
4-bit DAC
comparator
and2

```
mouse L: schSingleSelectPt()          M: schHiMousePopUp()          R: schZoomFit(1.0 0.9)
>
```

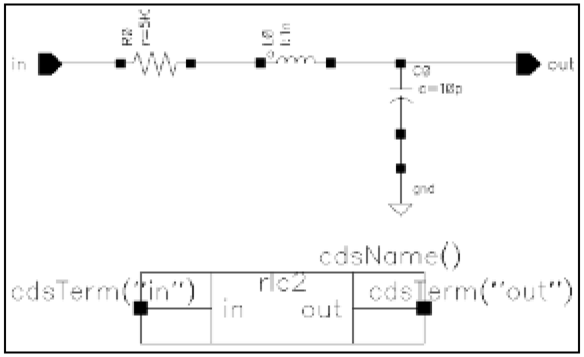
Verilog-A Modules in Schematic

Symbols with a behavioral view can be added into any schematic

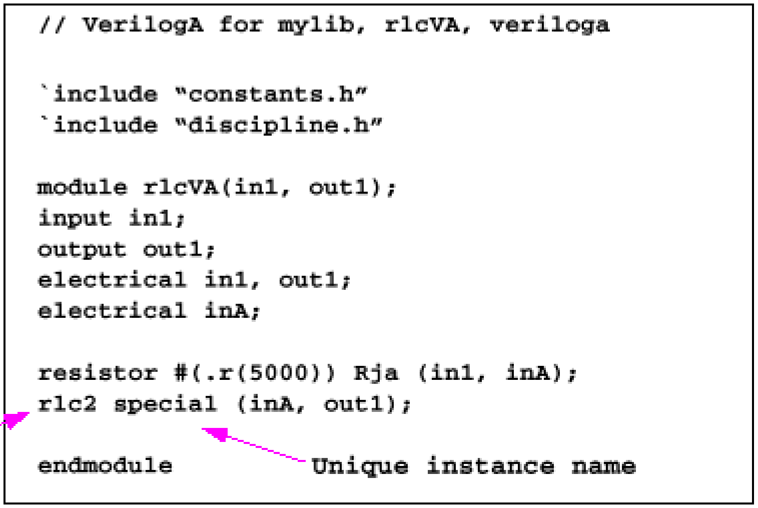


```
// VerilogA for mylib, rlc,
veriloga

`include “sonstants.h”
`include “discipline.h”

module rlc(in, out, ref) ;
...
```

Schematics in Verilog-A Modules



① A schematic and symbol view exist for *rlc2*.

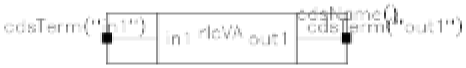


```
// VerilogA for mylib, rlcVA, veriloga

`include "constants.h"
`include "discipline.h"

module rlcVA(in1, out1);
input in1;
output out1;
electrical in1, out1;
electrical inA;

resistor #(.r(5000)) Rja (in1, inA);
rlc2 special (inA, out1);

endmodule
```

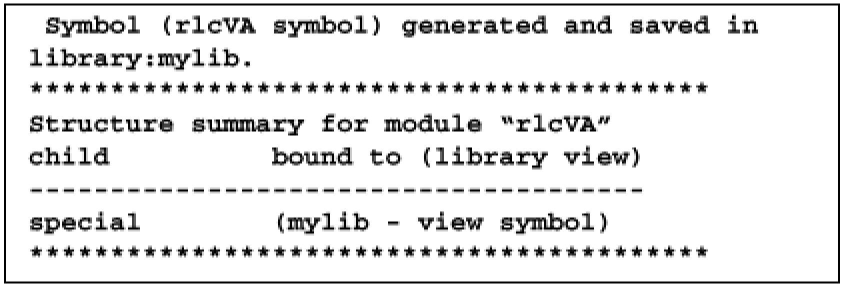
Unique instance name

Create instance of rlc2

② Create *veriloga* module for cell *rlcVA*.



③ Create a *symbol* for *rlcVA*.

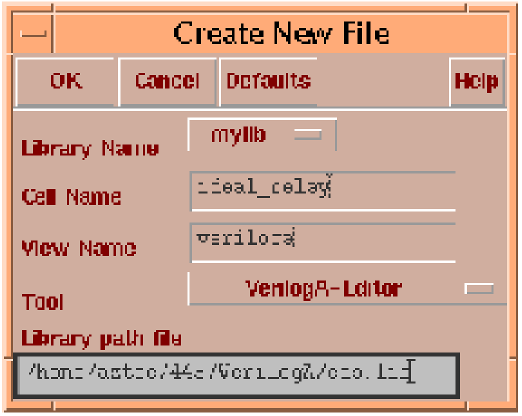


```
 Symbol (rlcVA symbol) generated and saved in
library:mylib.
***********************************************
Structure summary for module "rlcVA"
child            bound to (library view)
-----------------------------------------
special          (mylib - view symbol)
***********************************************
```

④ Note messages in the CIW.

Designing with Verilog-A in Analog Artist Environment

Verilog-A can be used in a standalone environment, in the Analog Workbench (AWB), or in the Analog Artist design environment.



Library Name: mylib

Cell Name: ideal_relay

View Name: veriloga

Tool: VerilogA-Editor

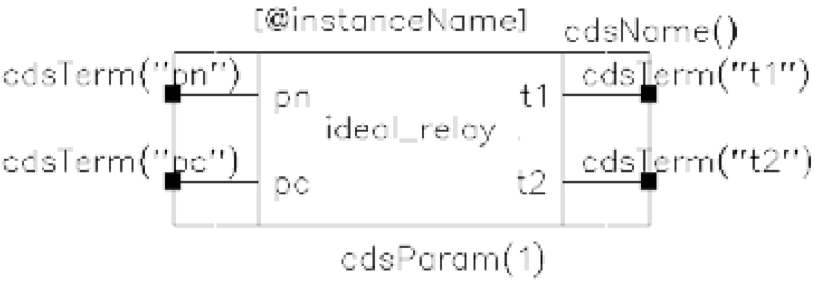
Library path file: /home/astro/44s/Verilog2/cds.lib

Create a *veriloga* view

Enter behavioral description

```
`include "discipline.h"
`include "constants.h"
module ideal_relay(pc, pn, t1, t2);
inout pc, pn, t1, t2;
electrical pc, pn, t1, t2;
parameter real thresh=2.5;
analog begin
            if( V(pc,pn) > thresh )
                  V(t1,t2) <+ 0.0;
            else
                  I(t1,t2) <+0.0;
      end
endmodule
```

Syntax is automatically checked when writing the file.

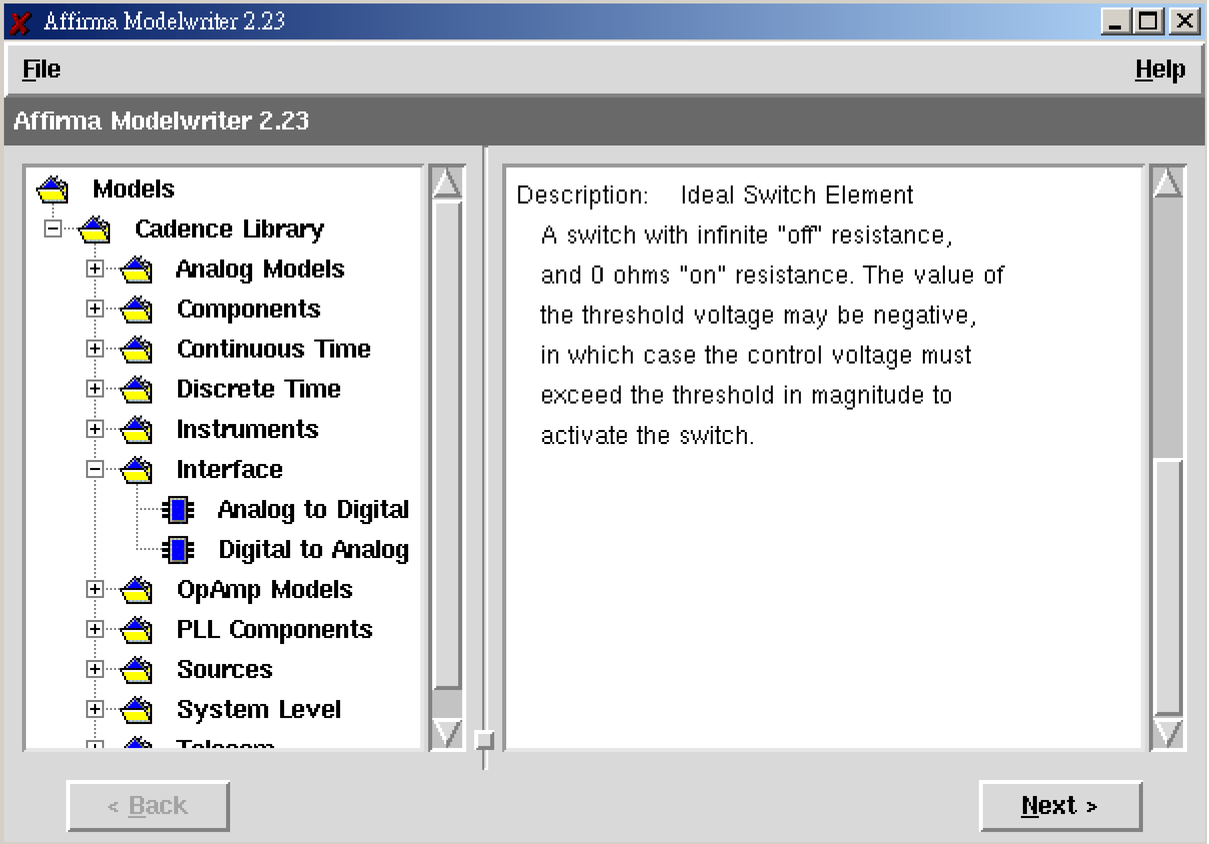


Create a symbol

Create Verilog-A Models by Modelwriter

Modelwrtier:
the utility for creating Verilog-A models for Cadence analog model library.

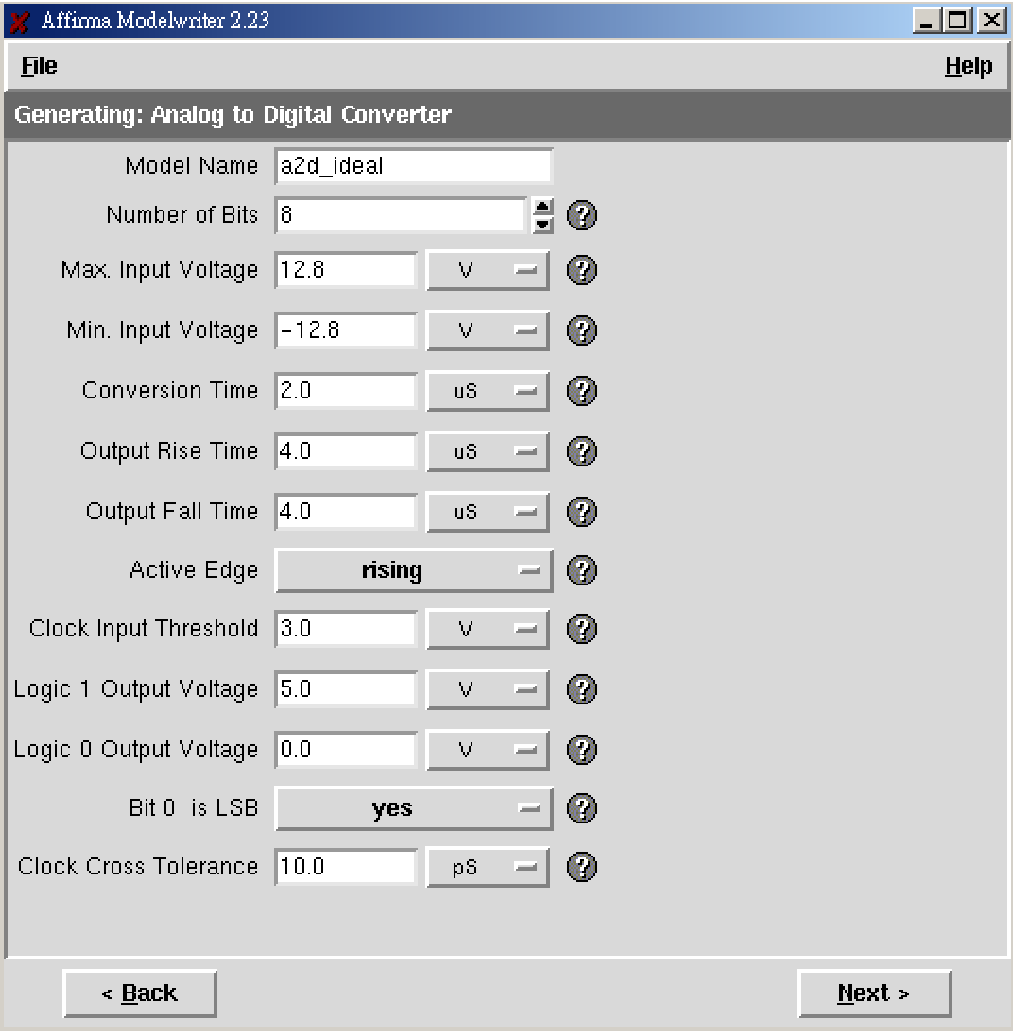
Use mouse select the desired component models then go to the next button.



Description: Ideal Switch Element
A switch with infinite "off" resistance,
and 0 ohms "on" resistance. The value of
the threshold voltage may be negative,
in which case the control voltage must
exceed the threshold in magnitude to
activate the switch.

Modelwriter Verilog-A Code

Use “Save Generated Code” to store the Verilog-A model or copy and paste to Verilog-A cellview.



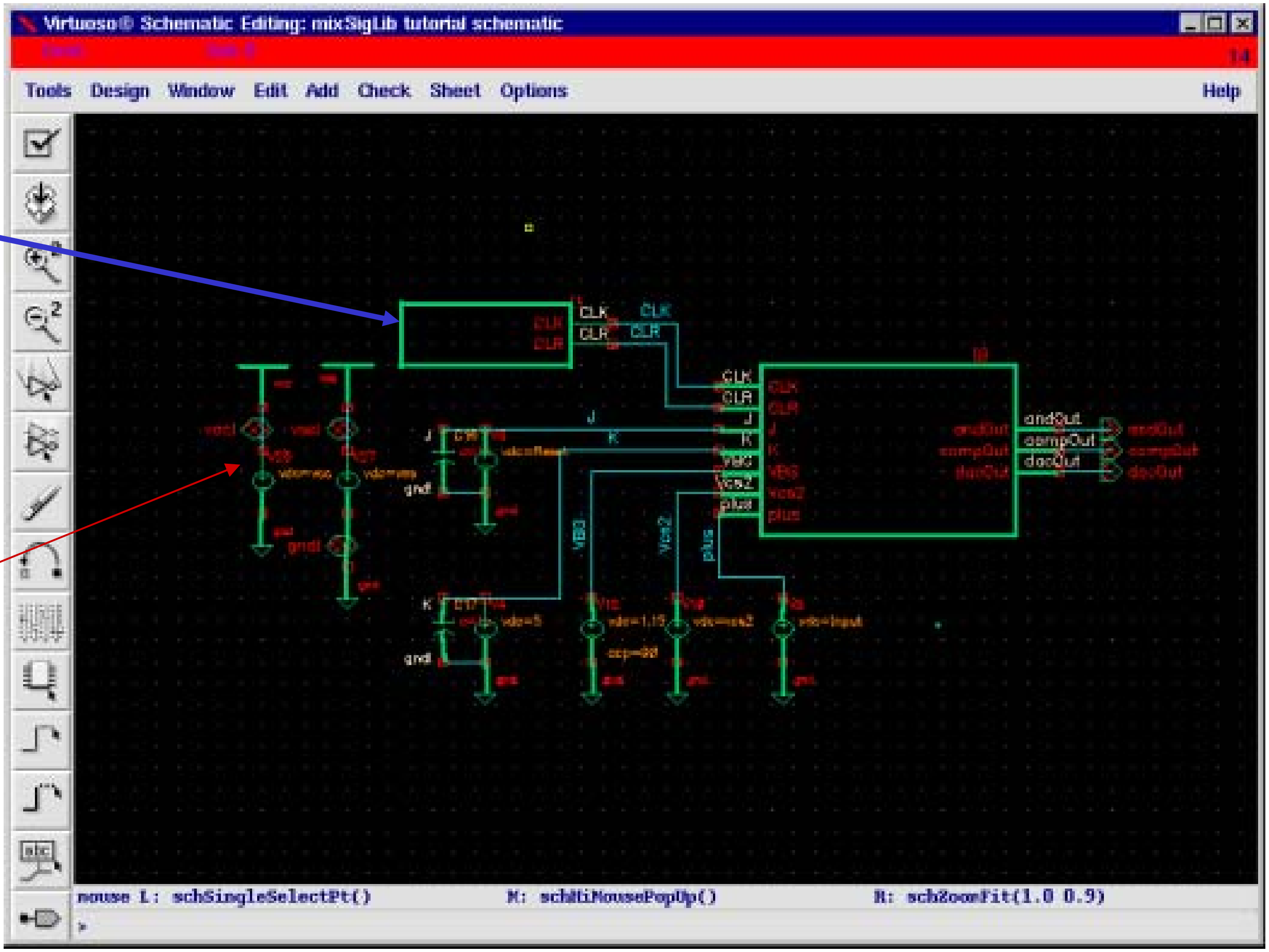
Generating: Analog to Digital Converter

Model Name	a2d_ideal	
Number of Bits	8	
Max. Input Voltage	12.8	V
Min. Input Voltage	-12.8	V
Conversion Time	2.0	uS
Output Rise Time	4.0	uS
Output Fall Time	4.0	uS
Active Edge	rising	
Clock Input Threshold	3.0	V
Logic 1 Output Voltage	5.0	V
Logic 0 Output Voltage	0.0	V
Bit 0 is LSB	yes	
Clock Cross Tolerance	10.0	pS

Adding Simulation Inputs

Digital Stimuli

Analog Stimuli

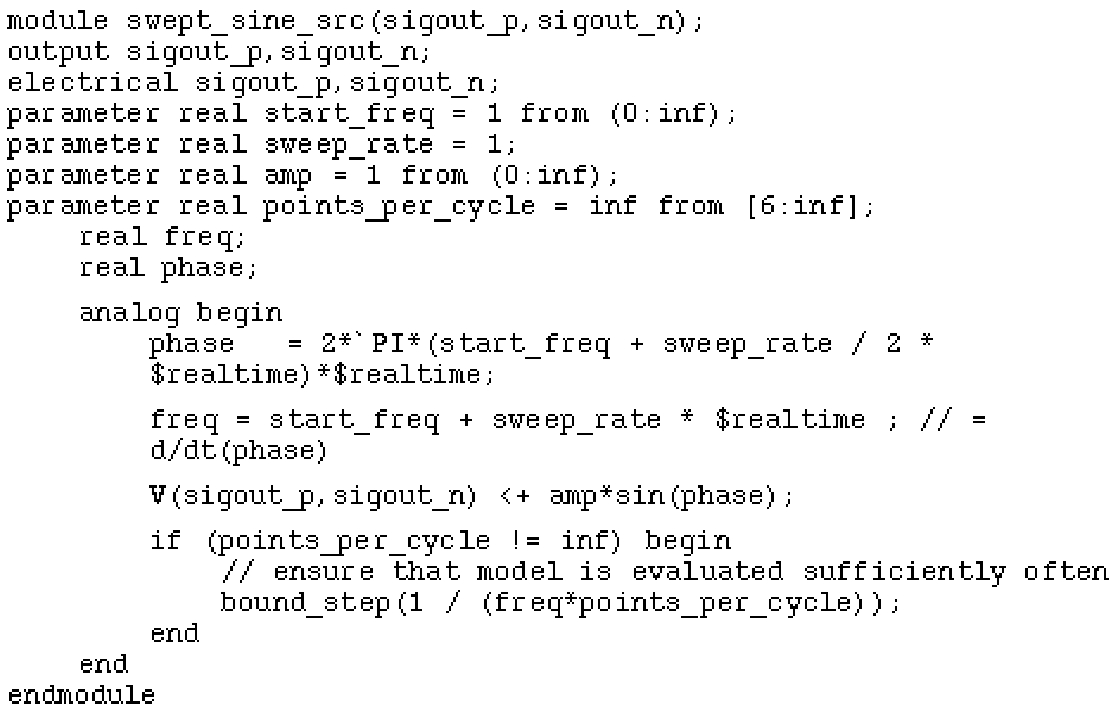

Virtuoso® Schematic Editing: mixSigLib tutorial schematic

mouse L: schSingleSelectPt()   M: schHiMousePopUp()   R: schZoomFit(1.0 0.9)

Analog Stimuli

The Analog Stimulus can be added either with circuit component in analogLib or with spectre AHDL stimulus format

- Edit a behavioral for the block
- create Symbol view
- Add the symbol in top schematic view



```
module swept_sine_src(sigout_p,sigout_n);
output sigout_p,sigout_n;
electrical sigout_p,sigout_n;
parameter real start_freq = 1 from (0:inf);
parameter real sweep_rate = 1;
parameter real amp = 1 from (0:inf);
parameter real points_per_cycle = inf from [6:inf];
     real freq;
     real phase;
     analog begin
          phase   = 2*`PI*(start_freq + sweep_rate / 2 *
          $realtime)*$realtime;
          freq = start_freq + sweep_rate * $realtime ; // =
          d/dt(phase)
          V(sigout_p,sigout_n) <+ amp*sin(phase);
          if (points_per_cycle != inf) begin
               // ensure that model is evaluated sufficiently often
               bound_step(1 / (freq*points_per_cycle));
          end
     end
endmodule
```

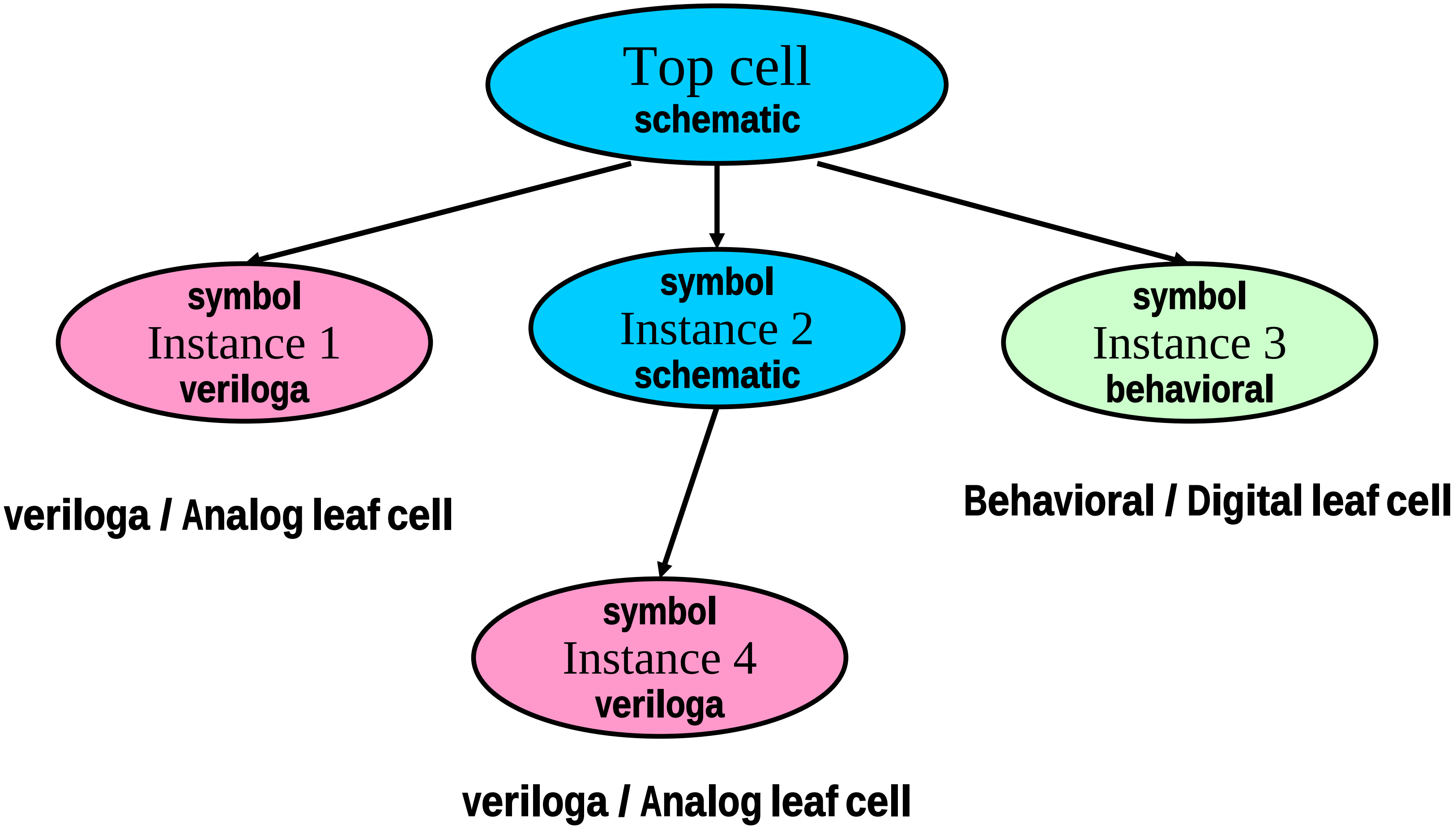
Digital Stimuli

1. Create a behavioral view for the stimulus block
2. Define the stimulus module
3. Add verilog command to force input signal
4. Create a symbol view for the block
5. Add the symbol into top schematic view

```
`timescale 10ns/10ns
//Define the stimulus block
module Stim(tx,precharge);
    output [1:16] tx ;
    output precharge ;
//Defines the registers
    reg [1:16] tx ;
    reg precharge ;
initial begin
   tx=16’h0000;
   precharge = 1’b0 ;
end
```

```
initial begin
    #2418 tx[3] = 1’b1 ;
    #17 tx[3] = 1’b0 ;
end
initial begin
    #1558 tx[6]=1’b1 ;
    #17 tx[6] = 1’b0 ;
initial begin
    #37 precharge = 1’b1 ;
    #11 precharge = 1’b0 ;
    #27 precharge = 1’b1 ;
end
```

Design Hierarchy



Top cell
schematic

symbol
Instance 1
veriloga

symbol
Instance 2
schematic

symbol
Instance 3
behavioral

symbol
Instance 4
veriloga

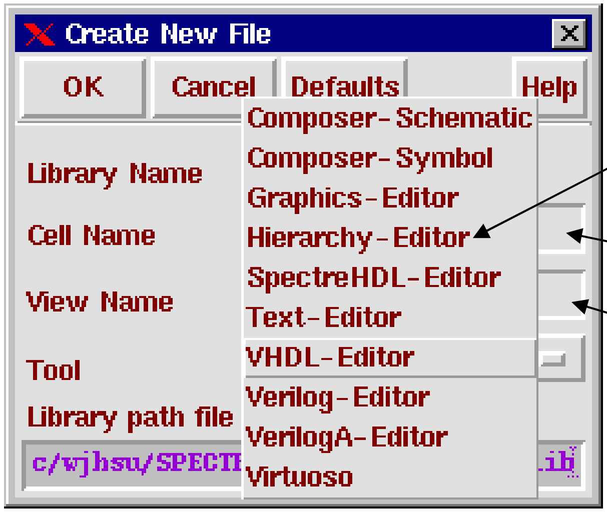
veriloga / Analog leaf cell

Behavioral / Digital leaf cell

veriloga / Analog leaf cell

Create Config View for Simulation

The mixed-signal simulation hierarchy is controlled by Hierarchy-Editor, which must be defined with **config** cell view

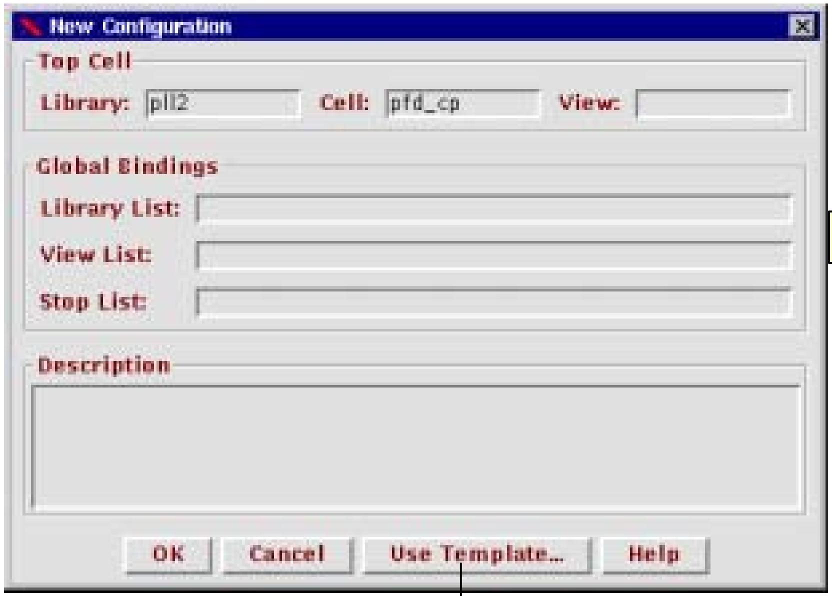


Use Create New File to create a new **config** view with Hierarchy-Editor

cell name is top circuit name for simulation

view name will be set as **config**

Set New Configuration



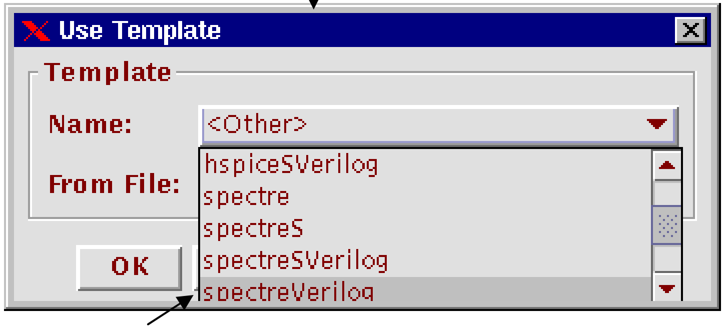
Library: pll2 Cell: pfd_cp View:

Library List:

View List:

Stop List:

1. Use Template sample information



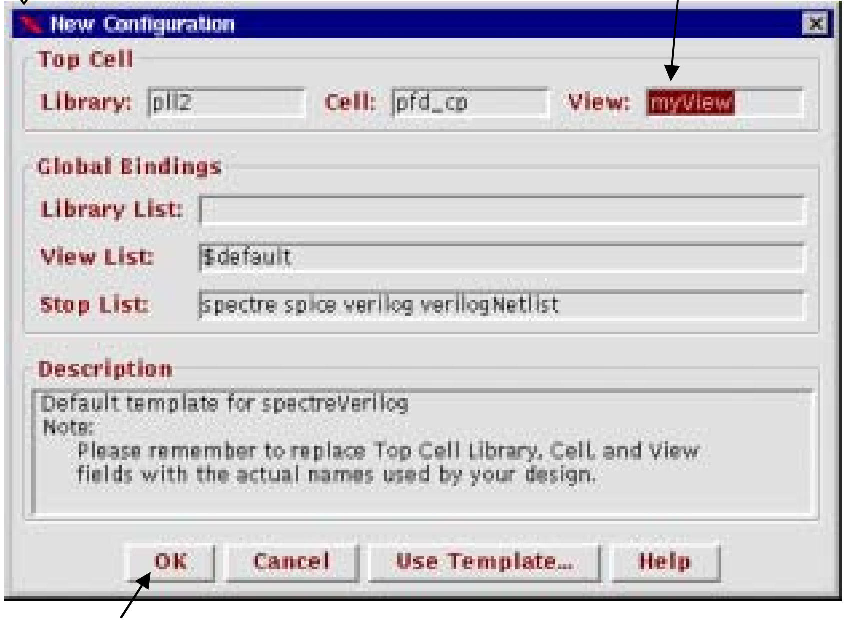
Name: <Other>

From File:

2. Change simulator to **spectreVerilog**

After template setting

3. Change the view name to **schematic** for simulation



Library: pll2 Cell: pfd_cp View: myView

Library List:

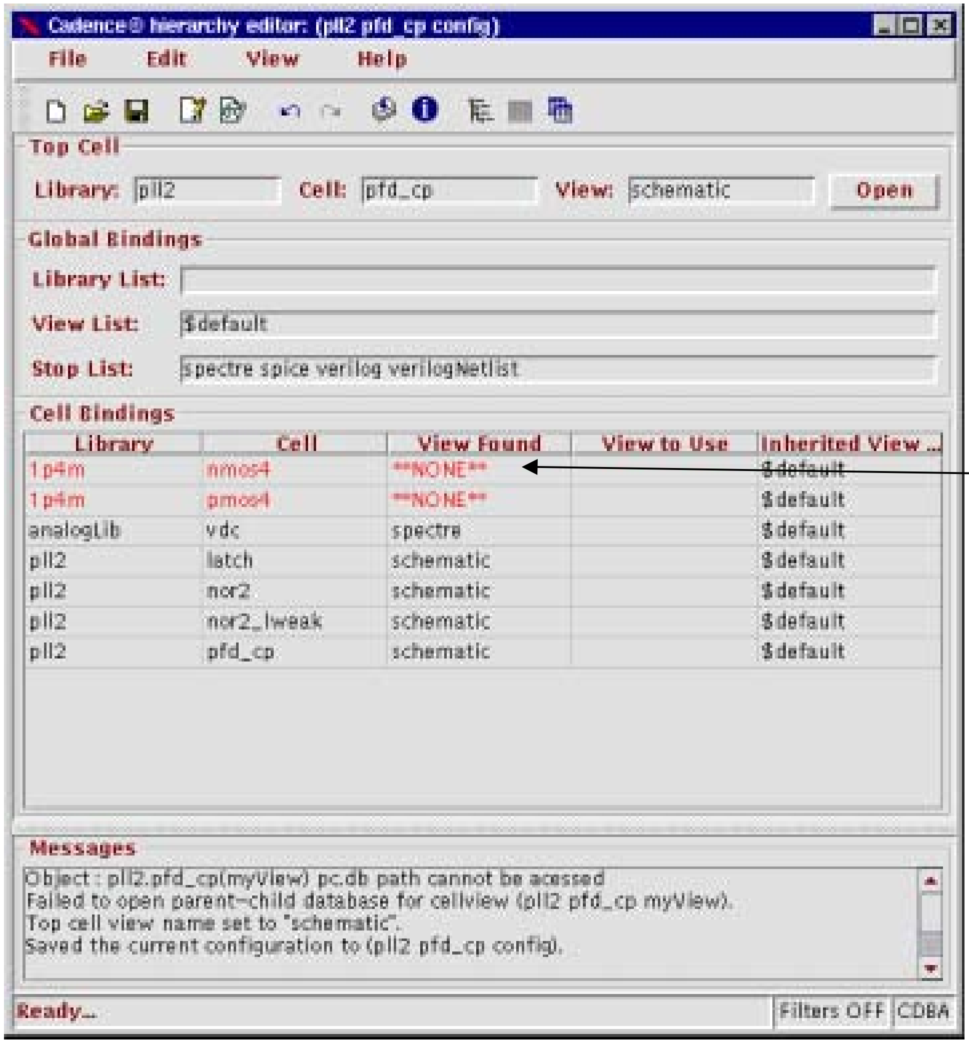
View List: $default

Stop List: spectre spice verilog verilogNetlist

Default template for spectreVerilog
Note:
Please remember to replace Top Cell Library, Cell, and View fields with the actual names used by your design.

4. Click OK

After Setting Configuration



Top Cell

Library: pll2 Cell: pfd_cp View: schematic

Global Bindings

Library List:

View List: $default

Stop List: spectre spice verilog verilogNetlist

Cell Bindings

Library	Cell	View Found	View to Use	Inherited View ...
1p4m	nmos4	**NONE**		$default
1p4m	pmos4	**NONE**		$default
analogLib	vdc	spectre		$default
pll2	latch	schematic		$default
pll2	nor2	schematic		$default
pll2	nor2_lweak	schematic		$default
pll2	pfd_cp	schematic		$default

Messages

```
Object : pll2.pfd_cp(myView) pc.db path cannot be acessed
Failed to open parent-child database for cellview (pll2 pfd_cp myView).
Top cell view name set to "schematic".
Saved the current configuration to (pll2 pfd_cp config).
```

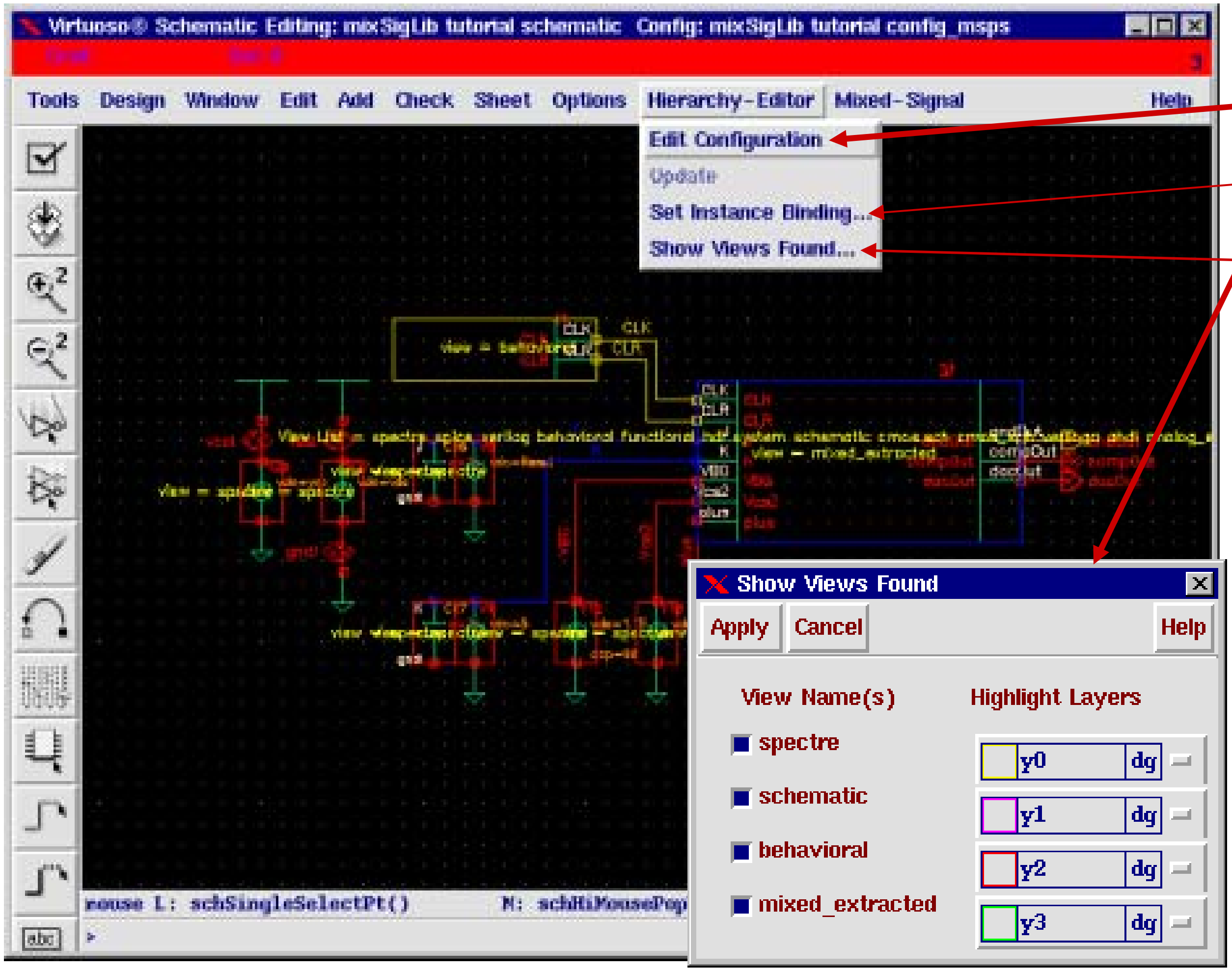
Ready... Filters OFF CDBA

The Hierarchy-Editor is shown, and all cells and views in the top cell will be listed.

The message shows that the cell used can not be identified because no available cell view was found

The problem must be cleared before simulation

Set Block Partition



```
mouse L: schSingleSelectPt()          M: schHiMousePop
```

View Name(s)	Highlight Layers	
■ spectre	y0	dg
■ schematic	y1	dg
■ behavioral	y2	dg
■ mixed_extracted	y3	dg

開啟 **hierarchy editor**

設定所使用的cell view

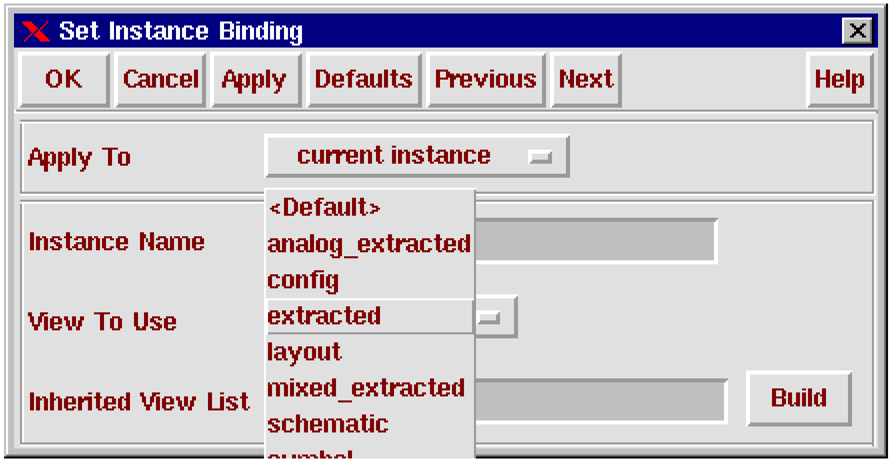
顯示 所用的**cell view,**
及其顏色設定

Schematic editor 中的 Hierarchy-Editor 及 Mixed-Signal 兩項menu係由選單 Tools->Mixed Signal Opts. 而產生。

Set Instance Binding

With menu **Hierarchy-Editor → Set Instance Binding…** , the following form will be shown for redefining the cell view to use for a certain instance.

Select a instance and choose the view to bin binded



Set Instance Binding

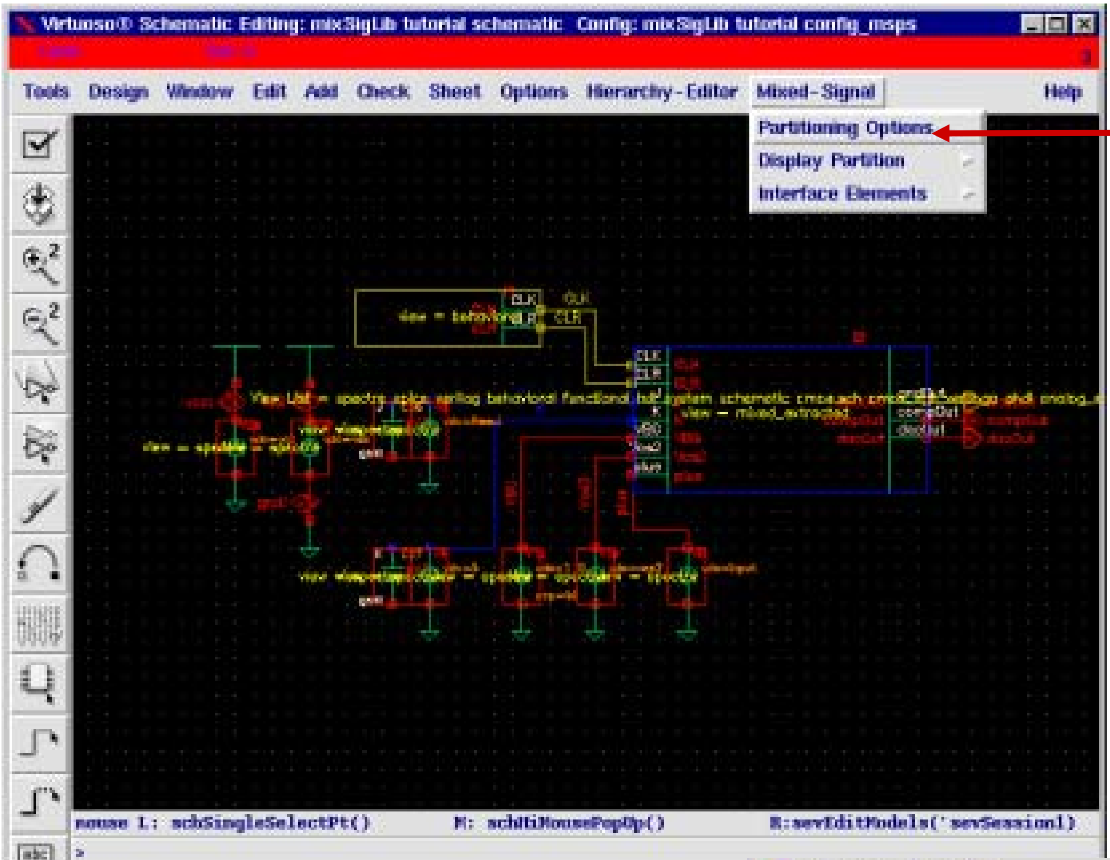
Apply To: current instance

Instance Name

View To Use: extracted

Inherited View List

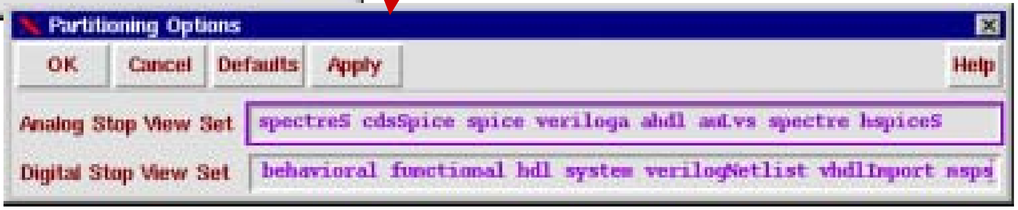
Check Block Partition



Virtuoso® Schematic Editing: mixSigLib tutorial schematic Config: mixSigLib tutorial config_msps

```
mouse L: schSingleSelectPt()        M: schHiMousePopUp()        R:sevEditModels('sevSession1)
```

Set the default cell view binding for partition



Partitioning Options

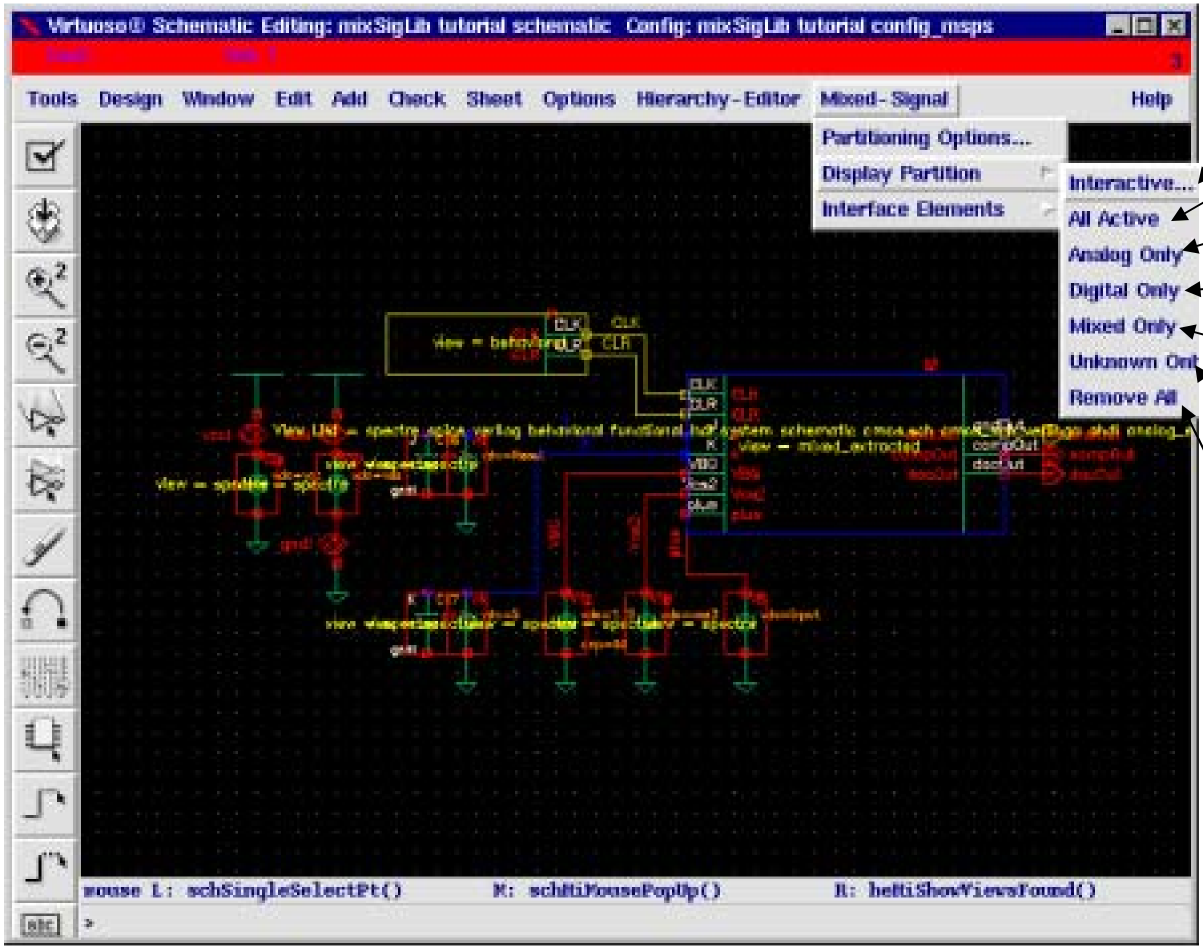
Analog Stop View Set: spectreS cdsSpice spice veriloga ahdl asl.vs spectre hspiceS

Digital Stop View Set: behavioral functional hdl system verilogNetlist vhdlImport msps

The default value can be preset in .cdsenv as
mmsimenv.conf digitalStopViewSet string “verilog ”
mmsimenv.conf analogStopViewSet string “spectre..”

Check Partition Results

With the Display Partition menu, the exact partition will be shown graphically.
The interface element will then be inserted between analog and digital blocks.



設定顯示的顏色及項目

顯示所有的區塊劃分結果

顯示類比電路區塊

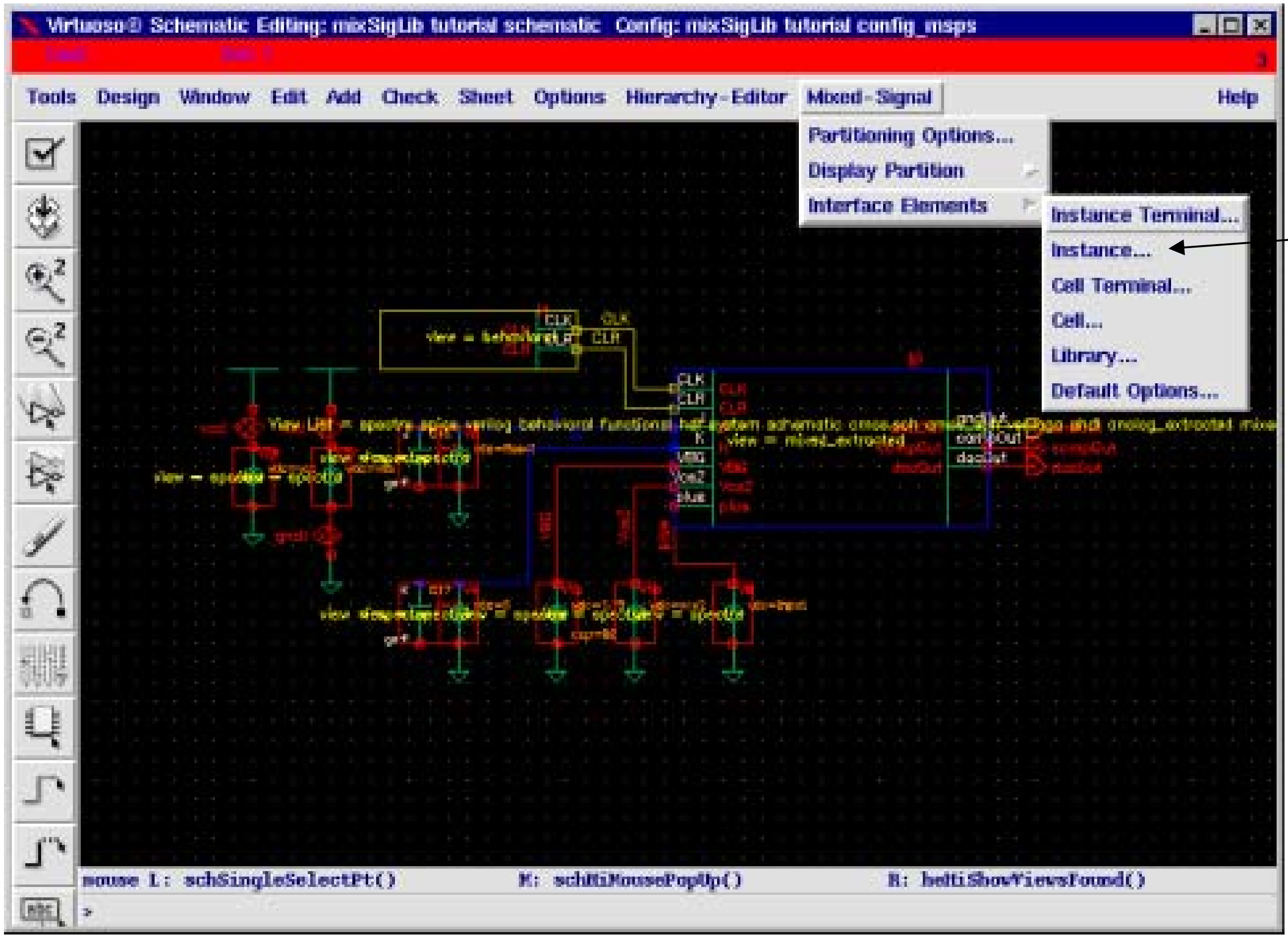
顯示數位電路區塊

顯示混合信號電路區塊
(即內同時包含數位及類比電路區塊)

顯示無法歸類之電路區塊

清除所有顯示的內容

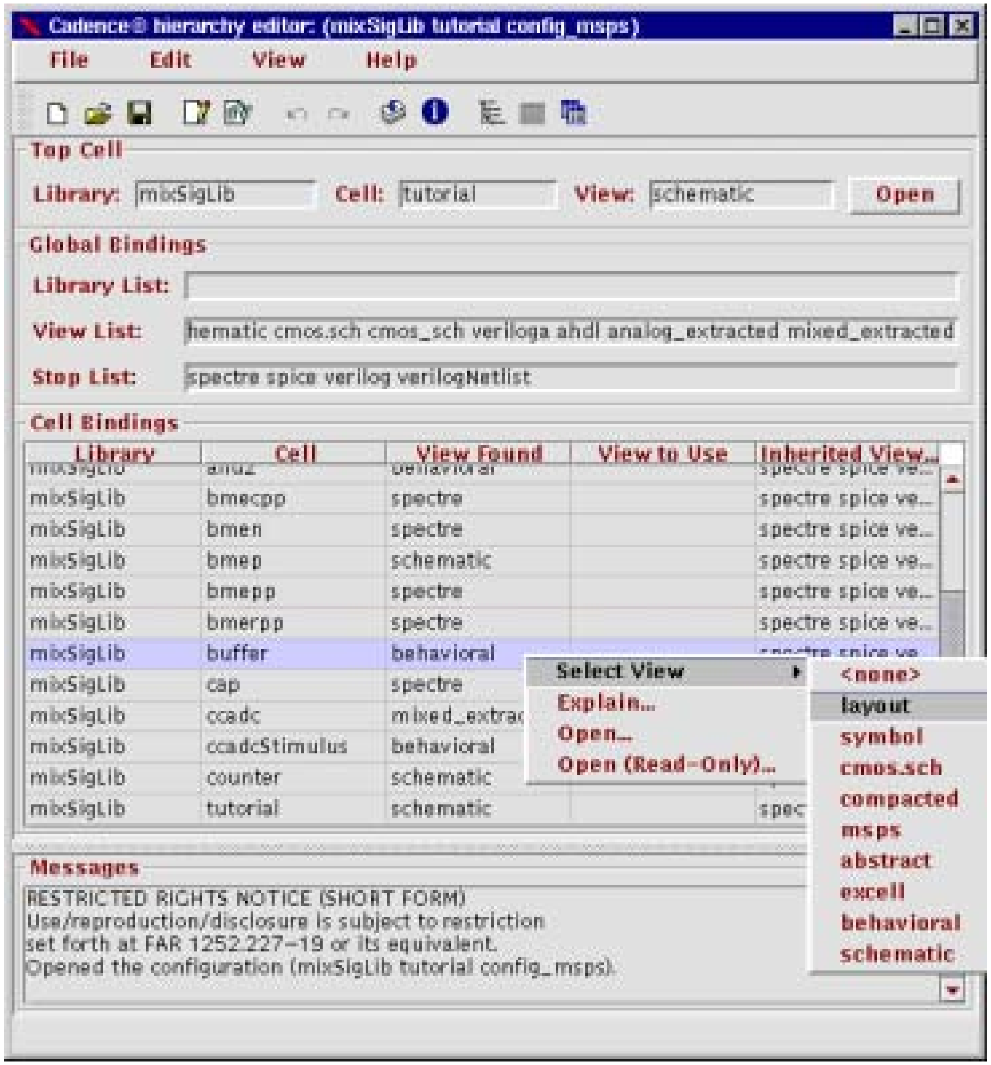
Define Detailed Interface Elements



Define the type of IE to be used

The type of IE can be defined by instance, cell or library

Change View Selection



Top Cell

Library: mixSigLib Cell: tutorial View: schematic Open

Global Bindings

Library List:

View List: hematic cmos.sch cmos_sch veriloga ahdl analog_extracted mixed_extracted

Stop List: spectre spice verilog verilogNetlist

Cell Bindings

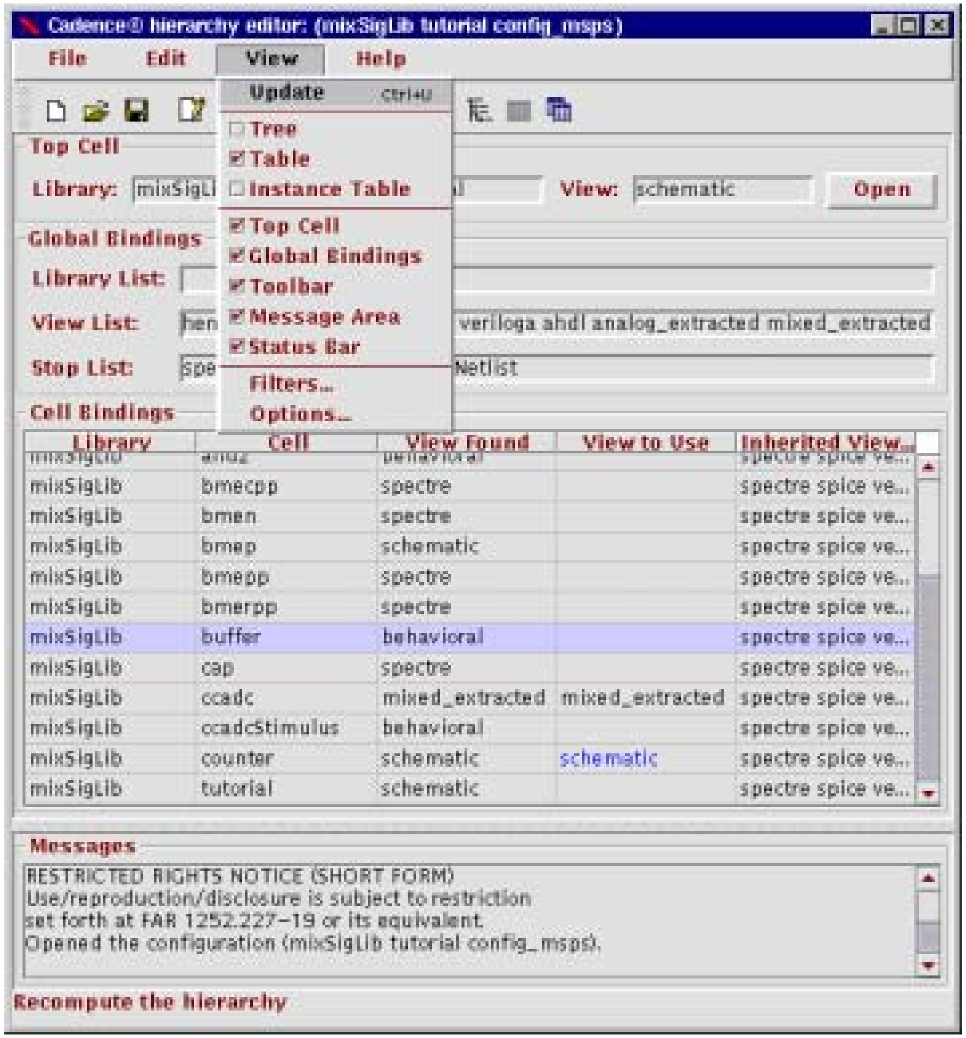
Library	Cell	View Found	View to Use	Inherited View...
mixSigLib	and2	behavioral		spectre spice ve...
mixSigLib	bmecpp	spectre		spectre spice ve...
mixSigLib	bmen	spectre		spectre spice ve...
mixSigLib	bmep	schematic		spectre spice ve...
mixSigLib	bmepp	spectre		spectre spice ve...
mixSigLib	bmerpp	spectre		spectre spice ve...
mixSigLib	buffer	behavioral		spectre spice ve
mixSigLib	cap	spectre		
mixSigLib	ccadc	mixed_extrac		
mixSigLib	ccadcStimulus	behavioral		
mixSigLib	counter	schematic		
mixSigLib	tutorial	schematic		spec

Messages

```
RESTRICTED RIGHTS NOTICE (SHORT FORM)
Use/reproduction/disclosure is subject to restriction
set forth at FAR 1252.227-19 or its equivalent.
Opened the configuration (mixSigLib tutorial config_msps).
```

With Hierarchy-Editor, cells that were used in the simulated cell will be listed. And each view used in simulation can be refined with the Editor.

Configuration of HE Display



Top Cell

Library: mixSigLi

View: schematic

Global Bindings

Library List:

View List: hen … veriloga ahdl analog_extracted mixed_extracted

Stop List: spe … Netlist

Cell Bindings

Library	Cell	View Found	View to Use	Inherited View...
mixSigLib	[illegible]	behavioral		spectre spice ve...
mixSigLib	bmecpp	spectre		spectre spice ve...
mixSigLib	bmen	spectre		spectre spice ve...
mixSigLib	bmep	schematic		spectre spice ve...
mixSigLib	bmepp	spectre		spectre spice ve...
mixSigLib	bmerpp	spectre		spectre spice ve...
mixSigLib	buffer	behavioral		spectre spice ve...
mixSigLib	cap	spectre		spectre spice ve...
mixSigLib	ccadc	mixed_extracted	mixed_extracted	spectre spice ve...
mixSigLib	ccadcStimulus	behavioral		spectre spice ve...
mixSigLib	counter	schematic	schematic	spectre spice ve...
mixSigLib	tutorial	schematic		spectre spice ve...

Messages

```
RESTRICTED RIGHTS NOTICE (SHORT FORM)
Use/reproduction/disclosure is subject to restriction
set forth at FAR 1252.227-19 or its equivalent.
Opened the configuration (mixSigLib tutorial config_msps).
```

Recompute the hierarchy

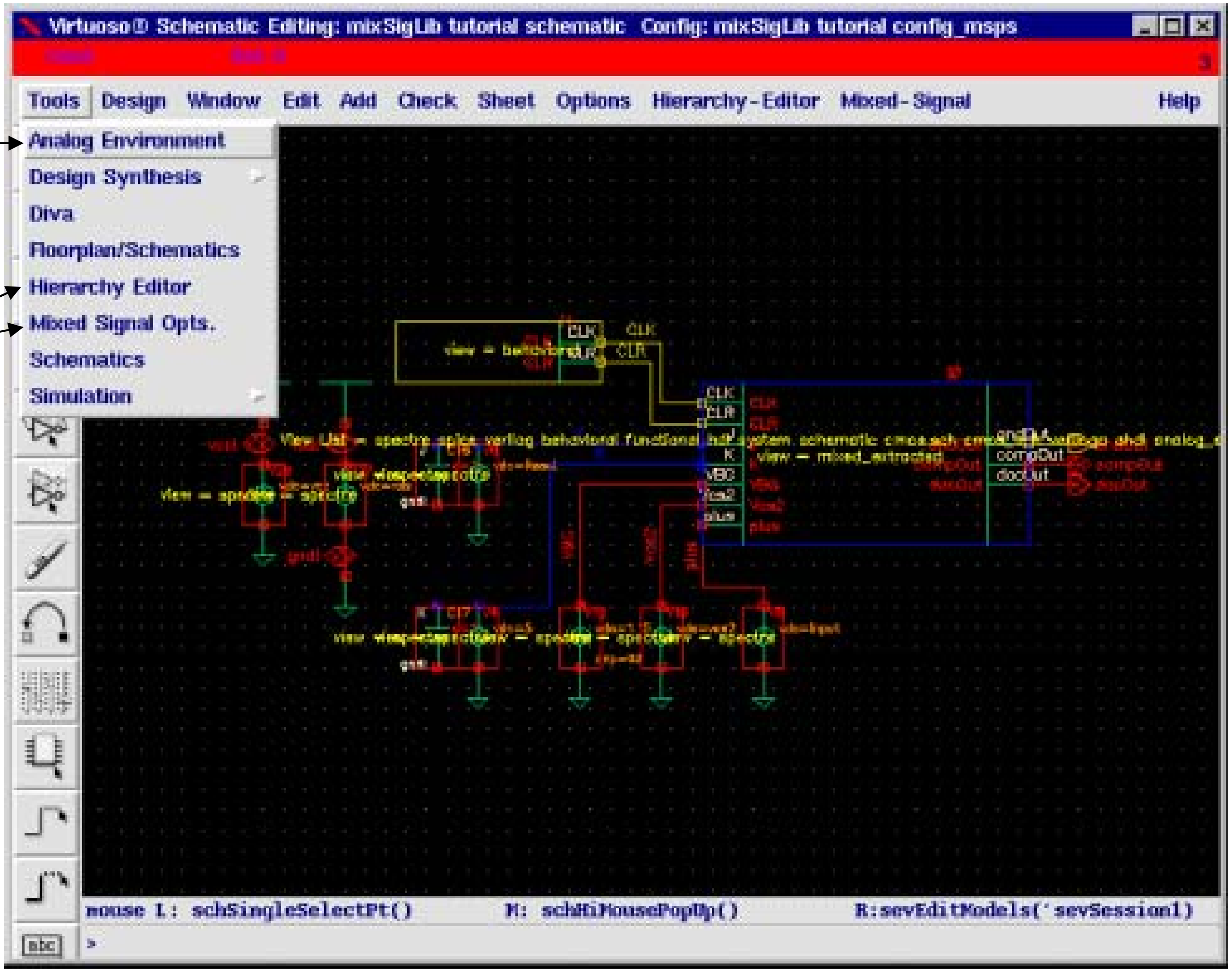
The contents to be shown in Hierarchy-Editor can be refined with this menu selection.

After refining the cell view to be used, use **Update** to change the simulation information.

Invoke Simulation Environment

開啟 Analog Artist Simulation 模擬環境

顯示這兩項menu



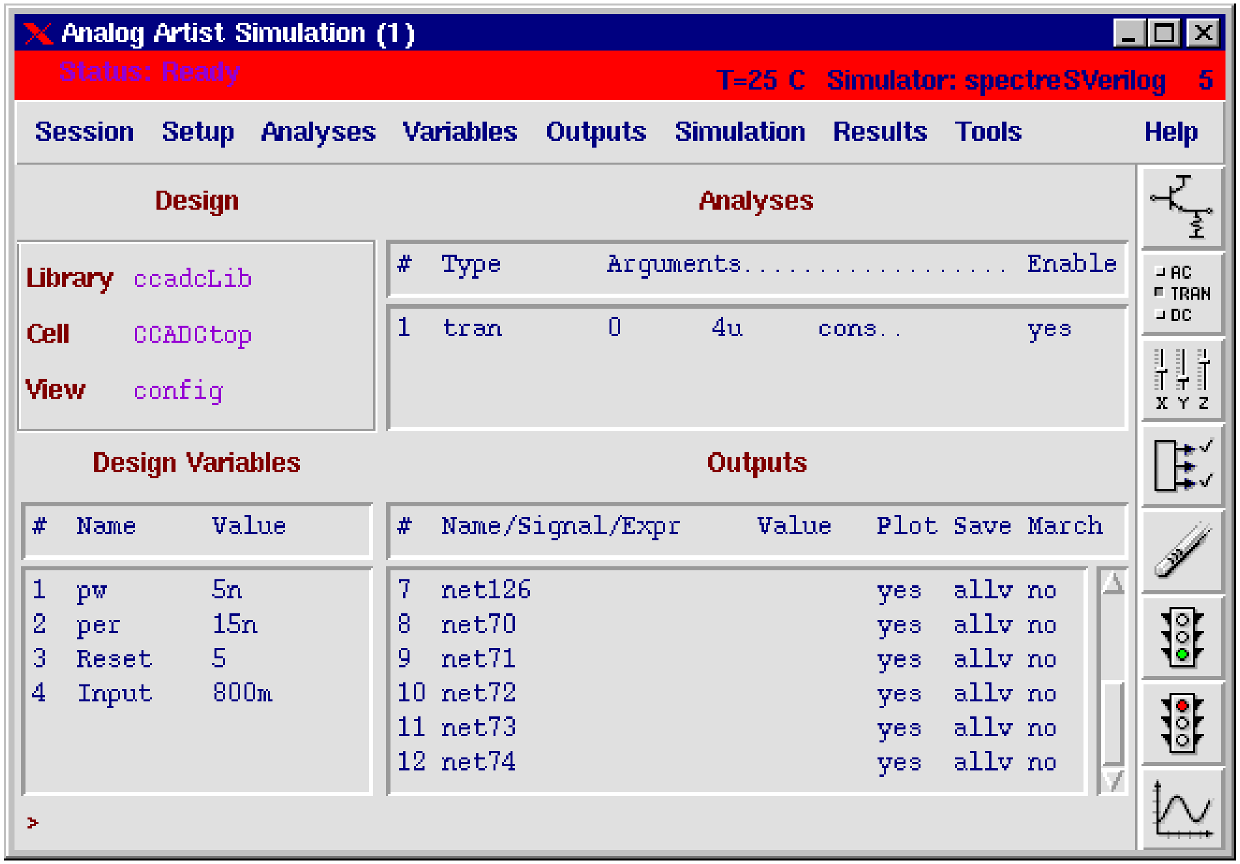
```
mouse L: schSingleSelectPt()          M: schHiMousePopUp()          R:sevEditModels('sevSession1)
>
```

Specify Simulation Environment

Use Analog Artist to control simulation progress

Select simulator, model path, include file, stimulus, analysis type

output saved/marched/plotted



Status: Ready

T=25 C Simulator: spectreSVerilog 5

Design

Library ccadcLib

Cell CCADCtop

View config

Analyses

#	Type	Arguments.................			Enable
1	tran	0	4u	cons..	yes

Design Variables

#	Name	Value
1	pw	5n
2	per	15n
3	Reset	5
4	Input	800m

Outputs

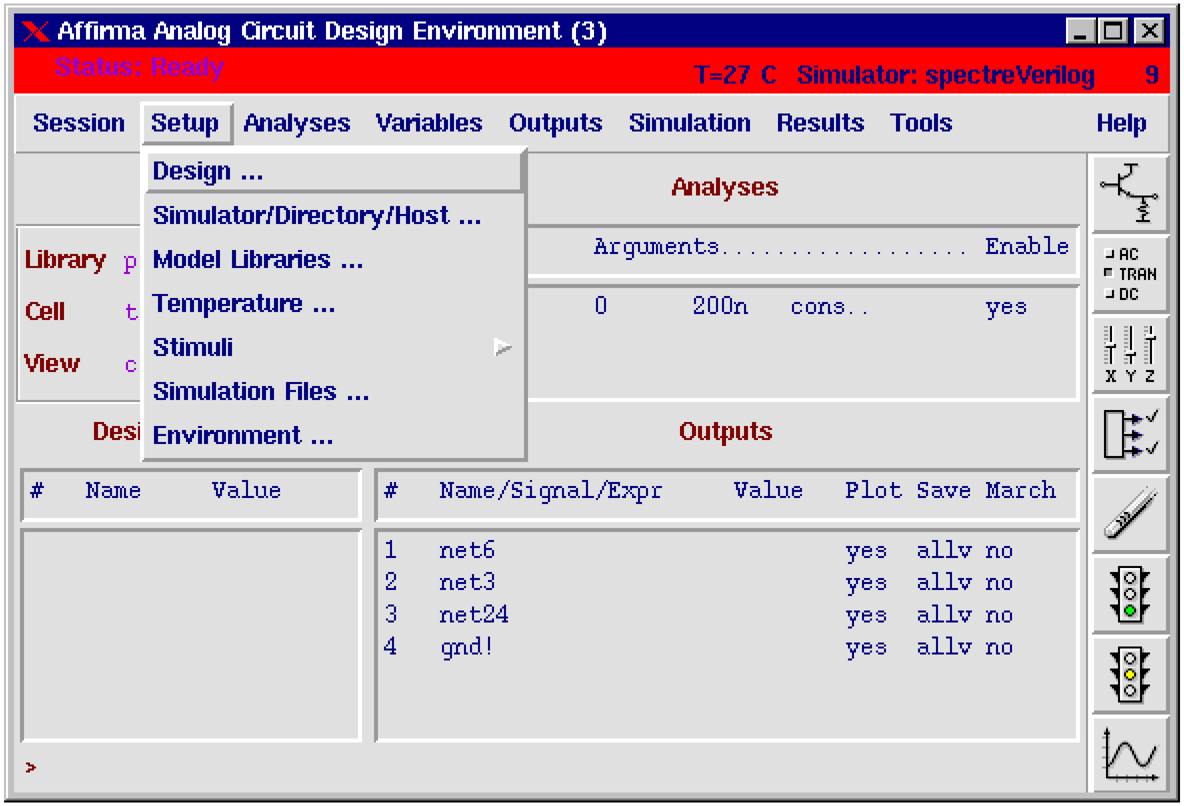
#	Name/Signal/Expr	Value	Plot	Save	March
7	net126		yes	allv	no
8	net70		yes	allv	no
9	net71		yes	allv	no
10	net72		yes	allv	no
11	net73		yes	allv	no
12	net74		yes	allv	no

>

Setup Menu in Analog Artist

With Setup window to define simulation initialization setup.

- Choose the simulator
- Define device model library
- Define temperature
- Define simulation inputs
- Define Verilog netlist options..



Analyses

Arguments.................			Enable
0	200n	cons..	yes

#	Name	Value

Outputs

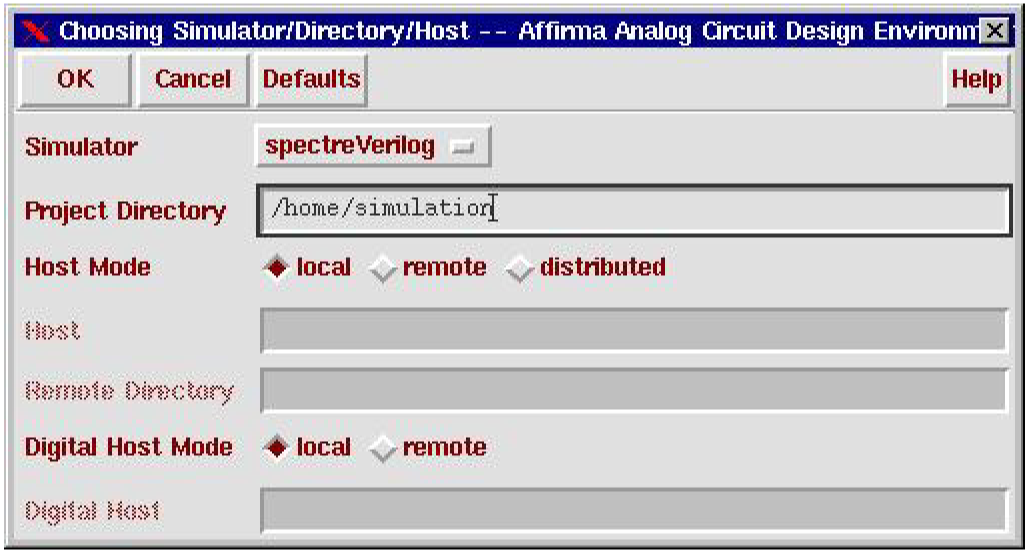
#	Name/Signal/Expr	Value	Plot	Save	March
1	net6		yes	allv	no
2	net3		yes	allv	no
3	net24		yes	allv	no
4	gnd!		yes	allv	no

Choosing Simulator/Directory/Host

Through **Setup → Simulator/Directory/Host**

Simulator :   S : socket

- cdsSpice
- hspiceS
- spectre
- spectreS
- cdsSpiceVerilog
- hspiceSVerilog
- spectreSVerilog
- spectreVerilog


Choosing Simulator/Directory/Host -- Affirma Analog Circuit Design Environm
OK
Cancel
Defaults
Help

Simulator: spectreVerilog

Project Directory: /home/simulation

Host Mode: ◆ local ◇ remote ◇ distributed

Host

Remote Directory

Digital Host Mode: ◆ local ◇ remote

Digital Host

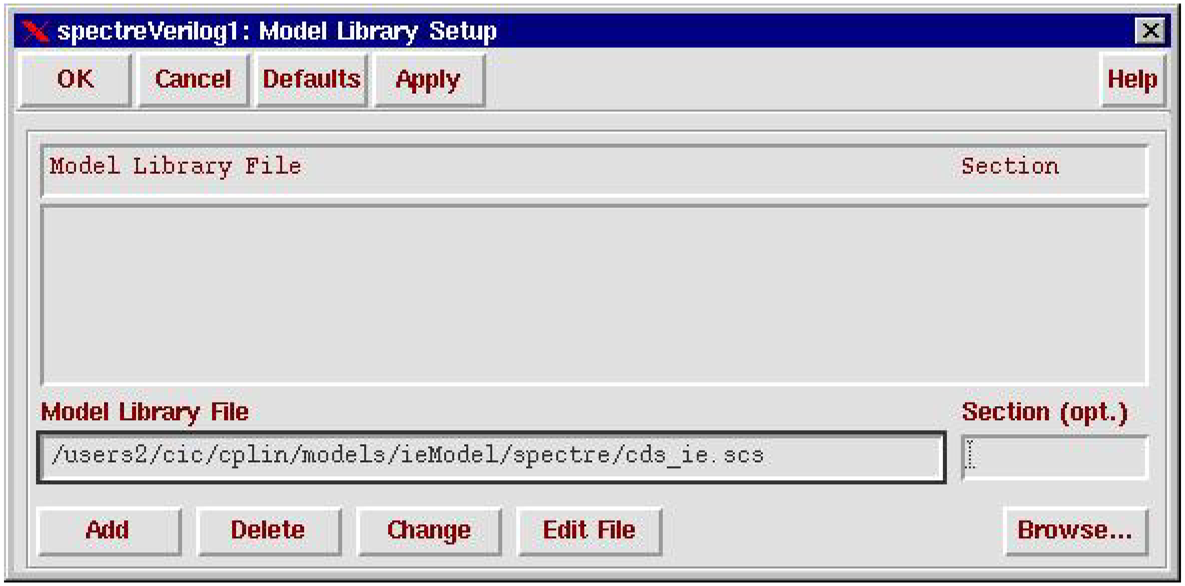
The exact simulation data directory is in
/home/simulation/***cellname/simulatorname/viewname***/netlist

Interface Element Model Definition

Use the menu ***Setup → Model Libraries*** to define the model contents of Interface elements, if any device model is also required, add with this also.

Example source of ieModels

/usr/cadence/IC/cur/tools/dfII/samples/artist/mixSig/ieModels/spectre
or /usr/cadence/IC/cur/tools/dfII/samples/artist/mixSig/ieModels/spectreS



Model Library File: /users2/cic/cplin/models/ieModel/spectre/cds_ie.scs

Section (opt.):

Digital to Analog Interface

For D → A interface, model the driving capability of digital parts

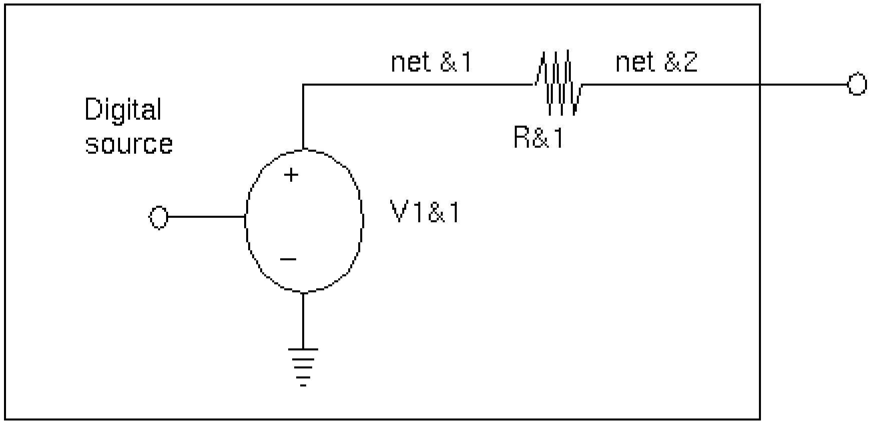
There are 3 levels of D → A interface model

Level 1 Digital to Analog Interface model

MOS1_d2a

Model Parameters

D2A_VL : input low voltage
D2A_VH : input high voltage
D2A_TR : rise time for low to high
D2A_TF :fall time for high to low
D2A_ROUT : Source resistance



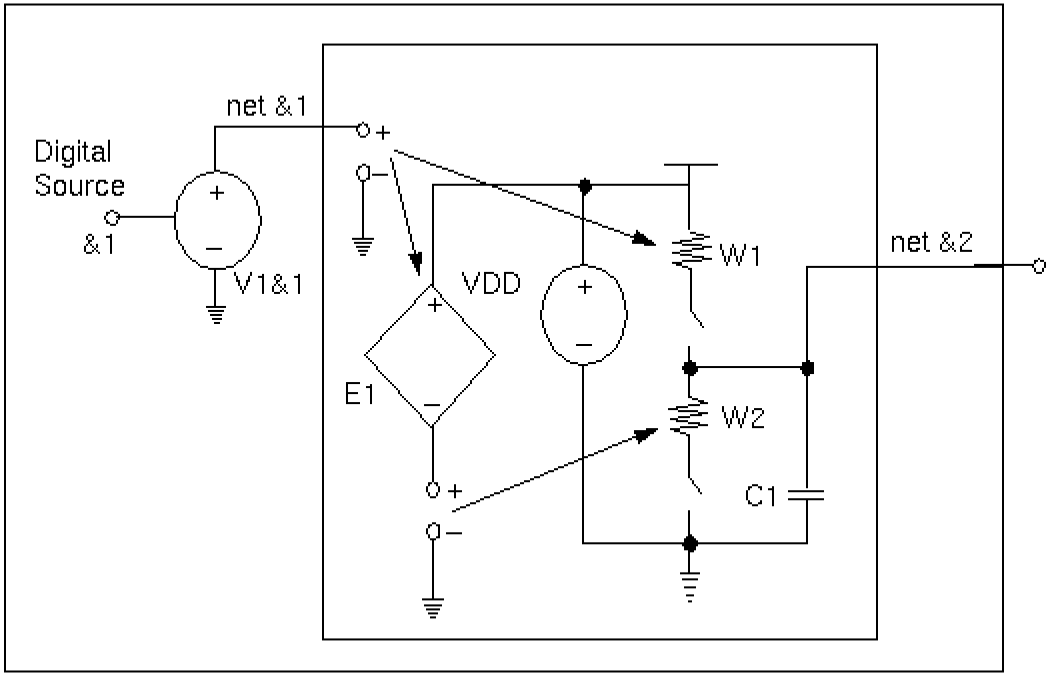
Digital to Analog Interface(Cont'd)

Level 2 Digital to Analog Interface Model
  - model the Z state , and independent sourcing, sinking

MOS2_d2a

Model parameters

D2A_VL
D2A_VH
D2A_TR
D2A_TF
D2A_HI_TH
D2A_LOW_TH
D2A_R1
D2A_R2



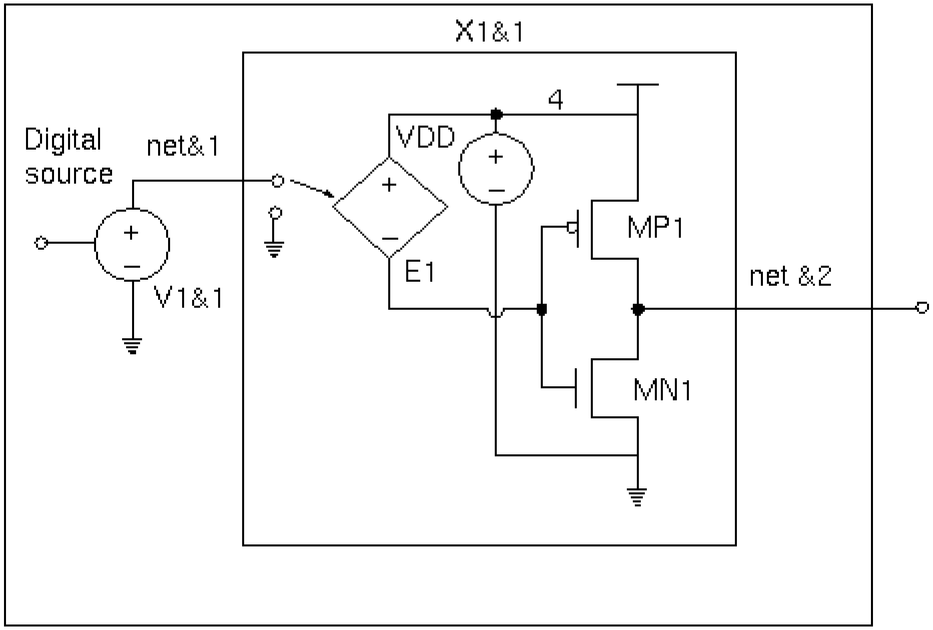
Digital to Analog Interface(Cont'd)

Level 3 Digital to Analog interface mode

MOS3_d2a

model parameters

D2A_VL
D2A_VH
D2A_TR
D2A_TF
D2A_TRANS_W : for NMOS



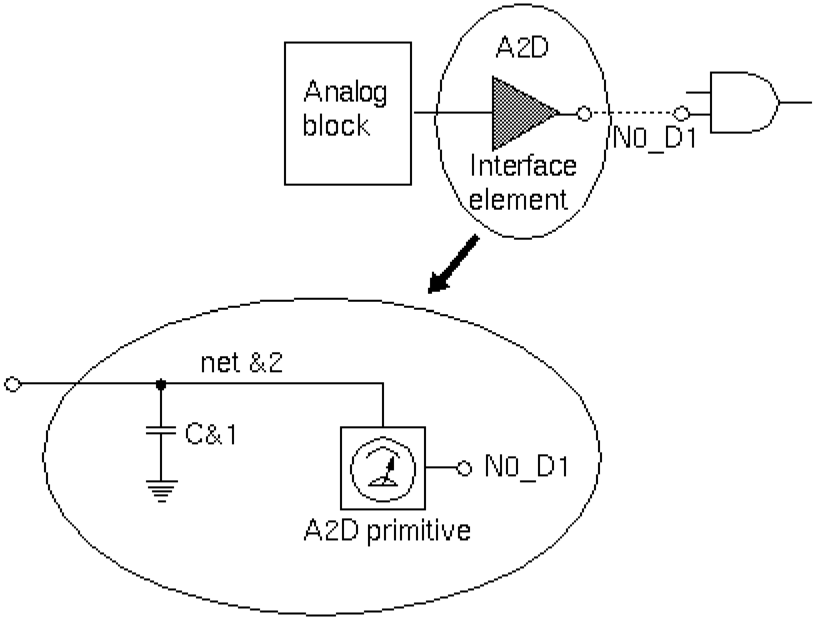
The PMOS width is 2*D2A_TRANS_W

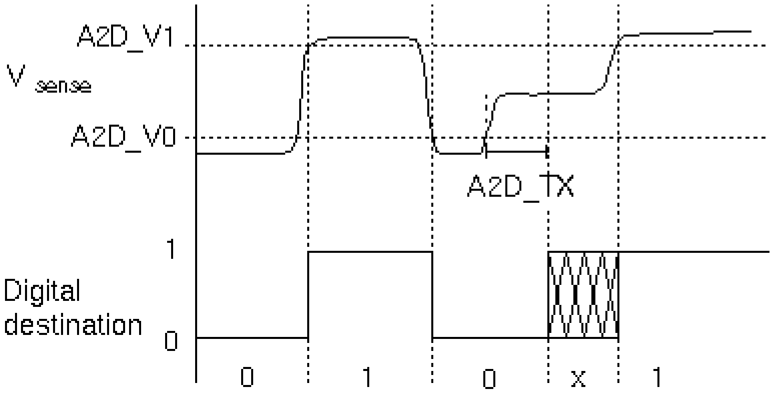
Analog to Digital Interface

Model parameters

MOS1_a2d

A2D_V0
A2D_V1
A2D_TX : voltage between V0 and V1 after TX will yield a logic X





Modeling the loading effect of digital nets to the analog parts

Modification of IE Parameters

- The instance level of IE parameters were defined by CDF parameters.

  Use menu Mixed-Signal → Interface Elements to redefine parameters

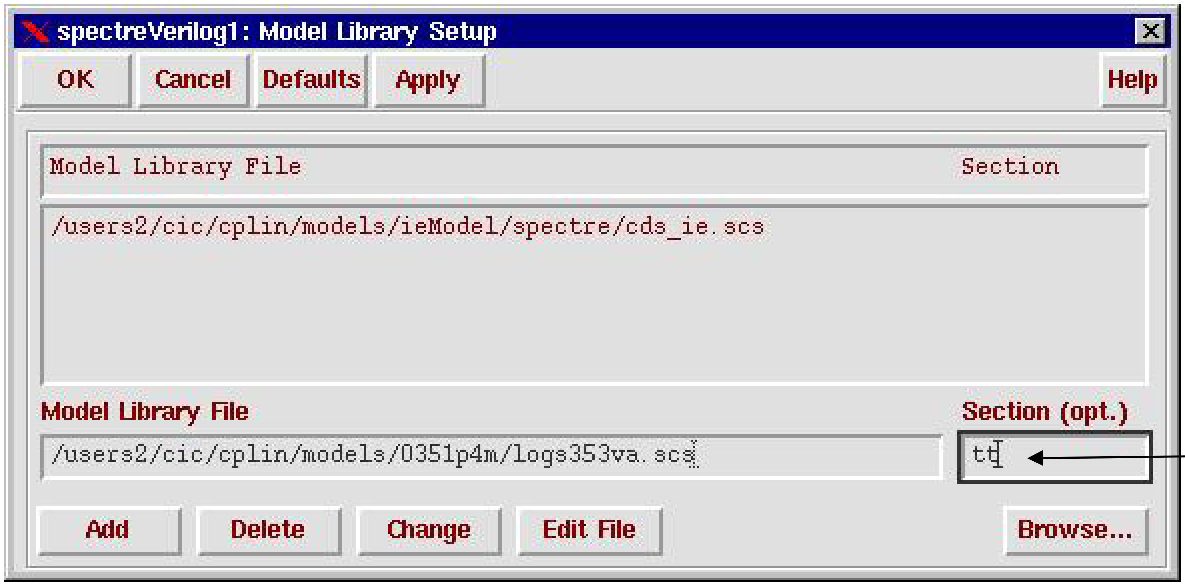
- The library level IE model is suggested to use MOS1
- The cell level of IE parameters were defined by the model definition. Copy the IE models into your directory, then modified as needed.

Device Model Specification

The active devices model for Spectre are provided with TSMC 0.35um, 0.25um,0.18um process.

All active devices used in schematic view of design must have corresponding model during simulation, such as NMOS,PMOS,DIO and BJTs. The device models might be provided with several files, specify those files in the following window.

The Section define the corner of model for simulation, must be consisted with model file.



Model Library File	Section
/users2/cic/cplin/models/ieModel/spectre/cds_ie.scs	

Model Library File: /users2/cic/cplin/models/0351p4m/logs353va.scs

Section (opt.): tt

Use tt corner

Device Model File Example

```
//      IN THIS MODEL LIB CONTAINS :
//      1.section tt (ss/ff /sf /fs)
//       ( 3.3V normal devices & 3.3V NMOS with ESD implant with different
//         geometric and corner models)
//      2.section bip
//       ( P+/NW/PSUB vertical PNP bipolar )
//      3.section dio
//       ( P+/NW, N+/PW & NW/PW diode )
//      4.section res
//       ( resistor model )
```

說明部份

```
section ss
parameters toxn=8.0e-9
parameters toxp=8.2e-9
parameters toxe=8e-09
parameters hdifp=3.8e-07
parameters hdife=9.55e-07
…………..
include "logs353va.scs" section=mos
endsection ss
```

Define the section name

Set Design Variables

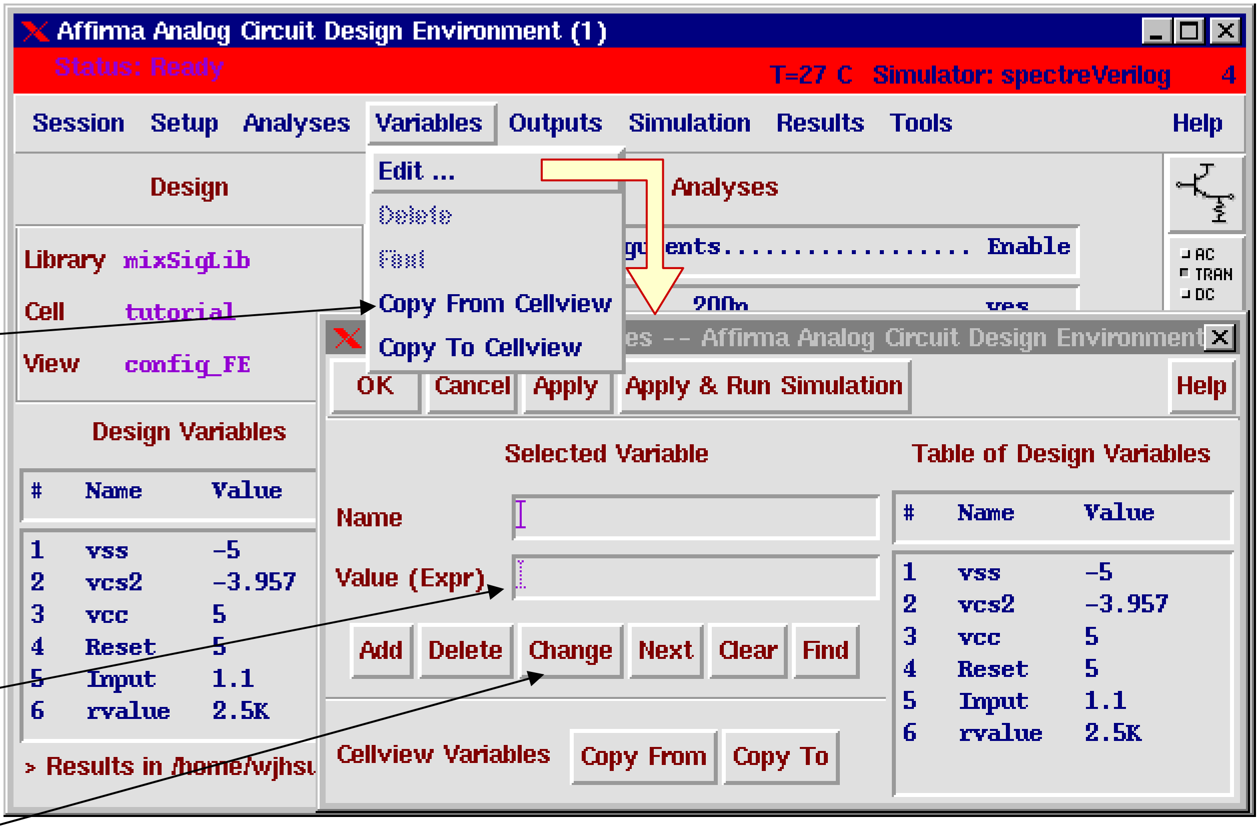
Some of the parameter values might be specified with a variable name. Then the value of variable can be given at this stage.

Get the lists of variables

Select the variable name with mouse in right Table

Specify the value

Press **Change** to update



Library mixSigLib
Cell tutorial
View config_FE

#	Name	Value
1	vss	-5
2	vcs2	-3.957
3	vcc	5
4	Reset	5
5	Input	1.1
6	rvalue	2.5K

Name

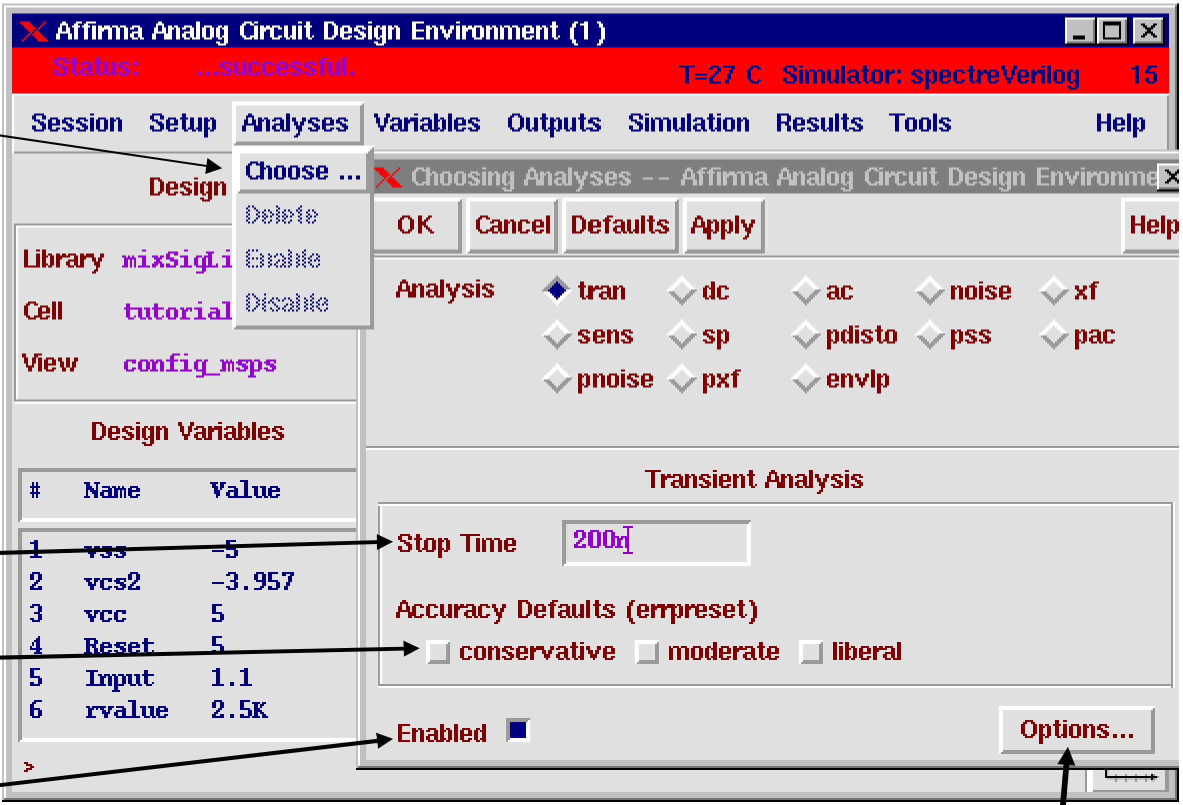
Value (Expr)

#	Name	Value
1	vss	-5
2	vcs2	-3.957
3	vcc	5
4	Reset	5
5	Input	1.1
6	rvalue	2.5K

Edit form of Variables

Choose Analysis Type

Define the analysis type through the window items



Affirma Analog Circuit Design Environment (1)

Status: ...successful. T=27 C Simulator: spectreVerilog 15

Library mixSigLi

Cell tutorial

View config_msps

Design Variables

#	Name	Value
1	vss	-5
2	vcs2	-3.957
3	vcc	5
4	Reset	5
5	Input	1.1
6	rvalue	2.5K

Choosing Analyses -- Affirma Analog Circuit Design Environme

Analysis ◆ tran ◇ dc ◇ ac ◇ noise ◇ xf ◇ sens ◇ sp ◇ pdisto ◇ pss ◇ pac ◇ pnoise ◇ pxf ◇ envlp

Transient Analysis

Stop Time 200n

Accuracy Defaults (errpreset) □ conservative □ moderate □ liberal

Enabled ■

Invoke the analysis setting window

For Mixed-Signal simulation, only **tran** is meaningful

Set the simulation time

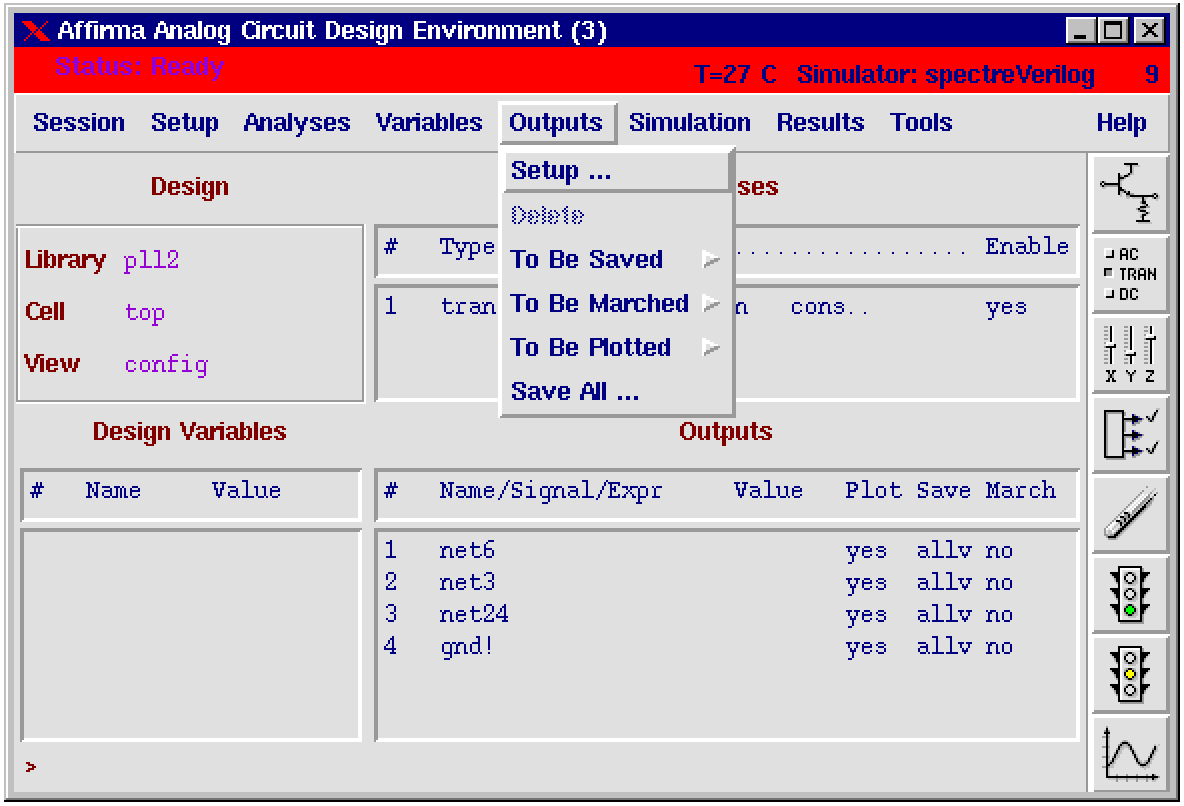
Set the accuracy flag

Check this box to enable this simulation

Set the transient simulation options
Choose compression within this options to reduce output file size

Select Output Nodes

Define the signals to plotted after simulation, the nodes can be selected within schematic window



Status: Ready

T=27 C Simulator: spectreVerilog 9

Design

Library pll2

Cell top

View config

ses

```
#   Type ................. Enable
1   tran  n   cons..        yes
```

Design Variables

#	Name	Value

Outputs

#	Name/Signal/Expr	Value	Plot	Save	March
1	net6		yes	allv	no
2	net3		yes	allv	no
3	net24		yes	allv	no
4	gnd!		yes	allv	no

>

Created Netlist(Analog)

HNL : Hierarchical Netlist
FNL : Flat netlist - for parasitic analysis, detailed IE mode

```
simulator lang= spectre
vi0 (5 0)  vsource type= pulse val0=-5.00000000E-01 val1=0.0 period=10.0
+delay=5e-9 rise=500e-12 fall=500e-12 width=1.0
vi1 (12 0)  vsource type= pulse val0=-5.00000000E-01 val1=0.0
+period=+1.50000000E-08 delay=1e-9 rise=500e-12 fall=500e-12
+width=+5.00000000E-09
qi2 (25 39 2)  tp1 region= fwd area=1 m=1.0

simulator lang= spice
* BEGIN Interface Element Header
da99978 99978 0 d2a src="99978" val0=500.0m val1=4.5 rise=1n fall=1n ron=1
R99978 99978 10 10
da99979 99979 0 d2a src="99979" val0=500.0m val1=4.5 rise=1n fall=1n ron=1
R99979 99979 14 10
da99980 99980 0 d2a src="99980" val0=500.0m val1=4.5 rise=1n fall=1n ron=1
R99980 99980 30 10
da99981 99981 0 d2a src="99981" val0=500.0m val1=4.5 rise=1n fall=1n ron=1
```

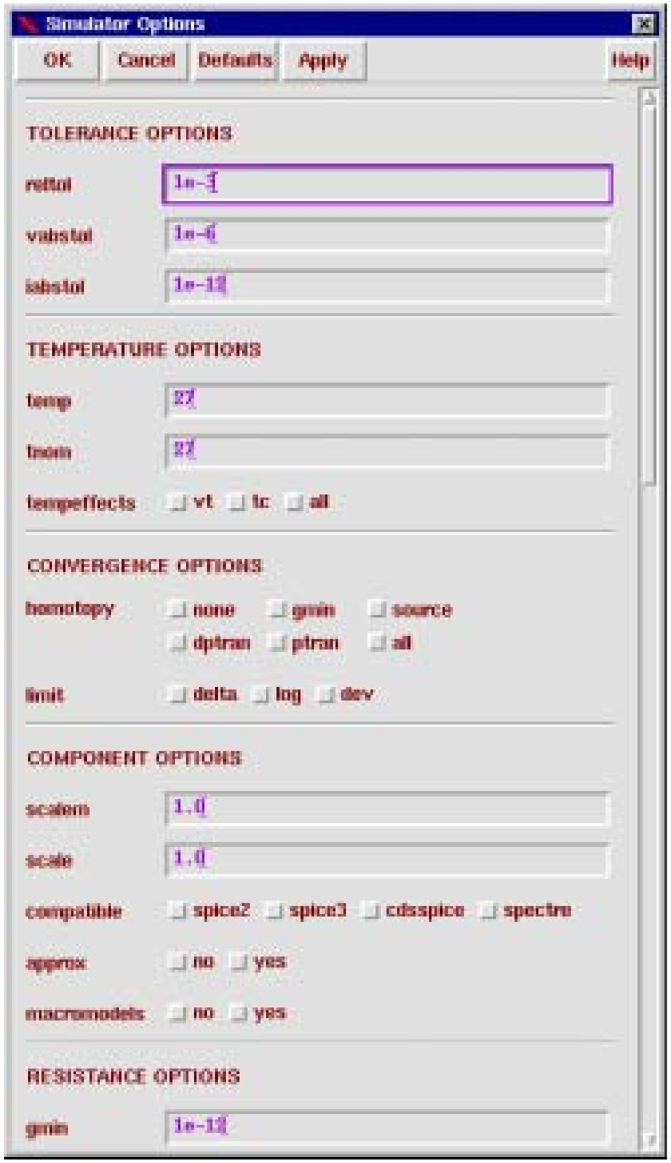
Created Netlist(Digital)

```
module CCADCtop;

supply1 N2;
supply0 N1;
supply0 N0;
// registers for ie elements
reg N12; // /net92
reg N7; // /comOut
reg N99995; // /net96
reg N99996; // /net79
reg N6; // /net53
reg N17; // /net96
reg N5; // /net79
specify
  specparam CDS_LIBNAME = "ccadcLib";
  specparam CDS_CELLNAME = "CCADCtop";
  specparam CDS_VIEWNAME = "schematic";
endspecify

buf I3( N8, N19 );
buf I4( N15, N28 );
```

```
// Begin Interface Element Header and Verimix
//Synchronization task
initial begin
     $vmx_initialize( "spectre", dc_mode_flag);
    $vmx_define_export( N10, "99978"); // /net70
     $vmx_define_import( N12, "99986"); // /net92
     $vmx_define_import( N7, "99987"); // /comOut
     $vmx_define_import( N99988, “99988”); // /net53
end
// End Interface Element Footer and Verimix
// Synchronization task

// Begin WSF Save Waveforms
initial begin
$save_waveform( "binary"
     ,"/net86", test.top.N4
     ,"/Q2", test.top.N23
);
end
// End WSF Save Waveforms
endmodule
```

Simulation Options for Analog Simulator

The analog simulator related control parameters can be specifed with this window.
The control options include
tolerance of solution
DC convergence solution method
Component format



TOLERANCE OPTIONS

reltol 1e-3

vabstol 1e-6

iabstol 1e-12

TEMPERATURE OPTIONS

temp 27

tnom 27

tempeffects □ vt □ tc □ all

CONVERGENCE OPTIONS

homotopy □ none □ gmin □ source
□ dptran □ ptran □ all

limit □ delta □ log □ dev

COMPONENT OPTIONS

scalem 1.0

scale 1.0

compatible □ spice2 □ spice3 □ cdsspice □ spectre

approx □ no □ yes

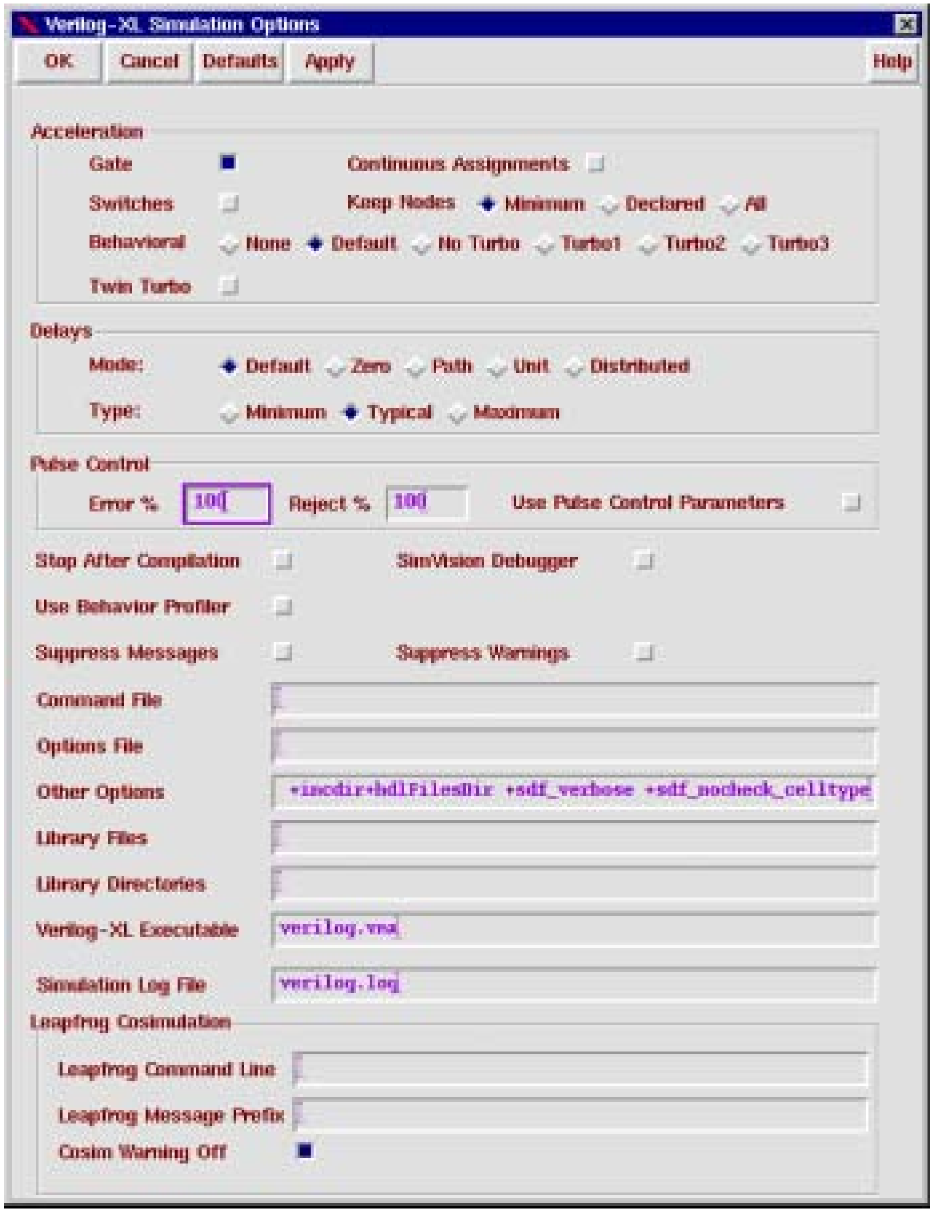
macromodels □ no □ yes

RESISTANCE OPTIONS

gmin 1e-12

Simulation Options for Digital Simulator

The options and environment setting that will be forwarded to verilog simulator can be defined with this window.



Acceleration

Gate ■ Continuous Assignments □

Switches □ Keep Nodes ● Minimum ○ Declared ○ All

Behavioral ○ None ● Default ○ No Turbo ○ Turbo1 ○ Turbo2 ○ Turbo3

Twin Turbo □

Delays

Mode: ● Default ○ Zero ○ Path ○ Unit ○ Distributed

Type: ○ Minimum ● Typical ○ Maximum

Pulse Control

Error % 100 Reject % 100 Use Pulse Control Parameters □

Stop After Compilation □ SimVision Debugger □

Use Behavior Profiler □

Suppress Messages □ Suppress Warnings □

Command File

Options File

Other Options +incdir+hdlFilesDir +sdf_verbose +sdf_nocheck_celltype

Library Files

Library Directories

Verilog-XL Executable verilog.vma

Simulation Log File verilog.log

Leapfrog Cosimulation

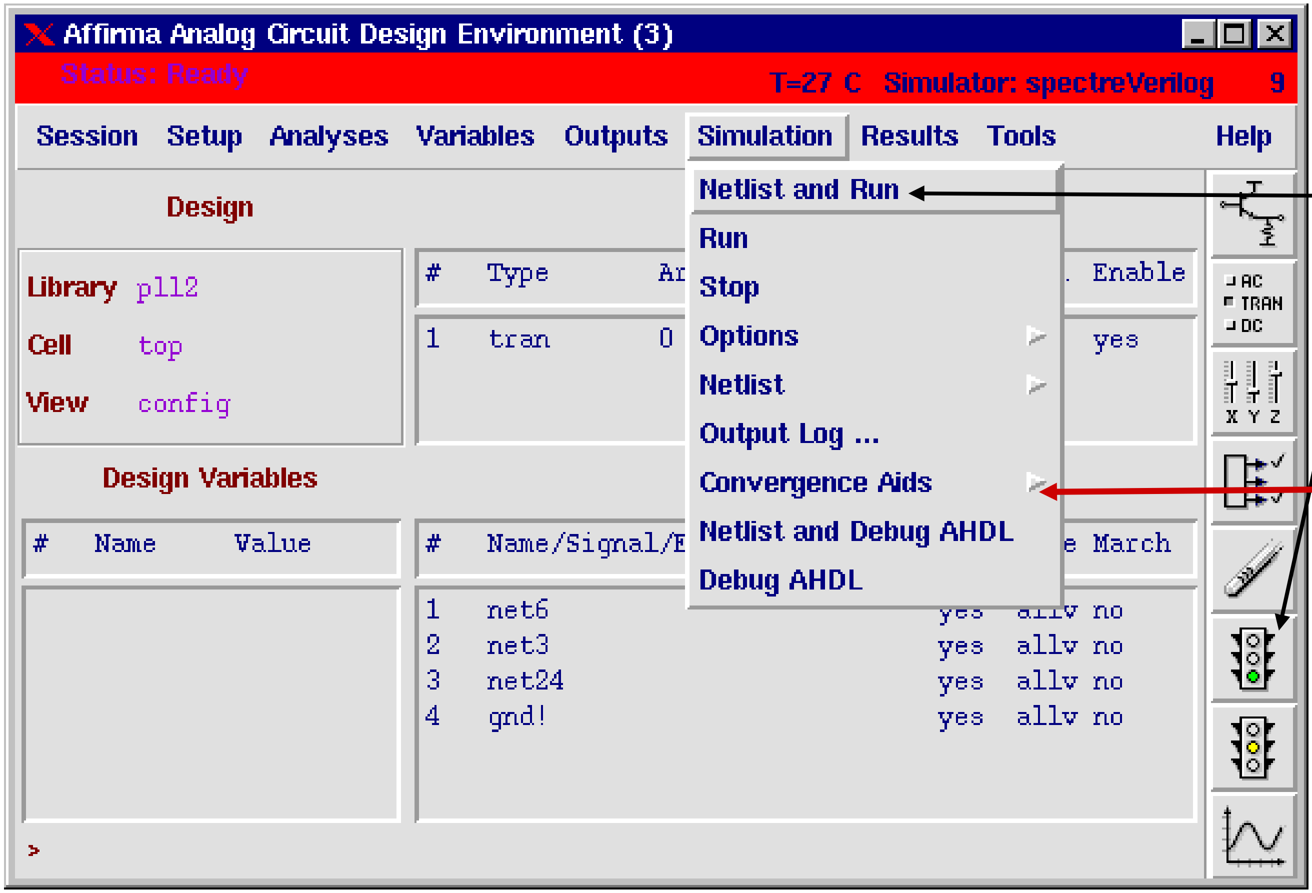
Leapfrog Command Line

Leapfrog Message Prefix

Cosim Warning Off ■

Submit the Simulation

Execute the simulation job with Run, or create the netlist with Netlist



Status: Ready T=27 C Simulator: spectreVerilog 9

Design

Library pll2

Cell top

View config

#	Type	Ar	Enable
1	tran	0	yes

Design Variables

#	Name	Value

#	Name/Signal/E			e	March
1	net6		yes	allv	no
2	net3		yes	allv	no
3	net24		yes	allv	no
4	gnd!		yes	allv	no

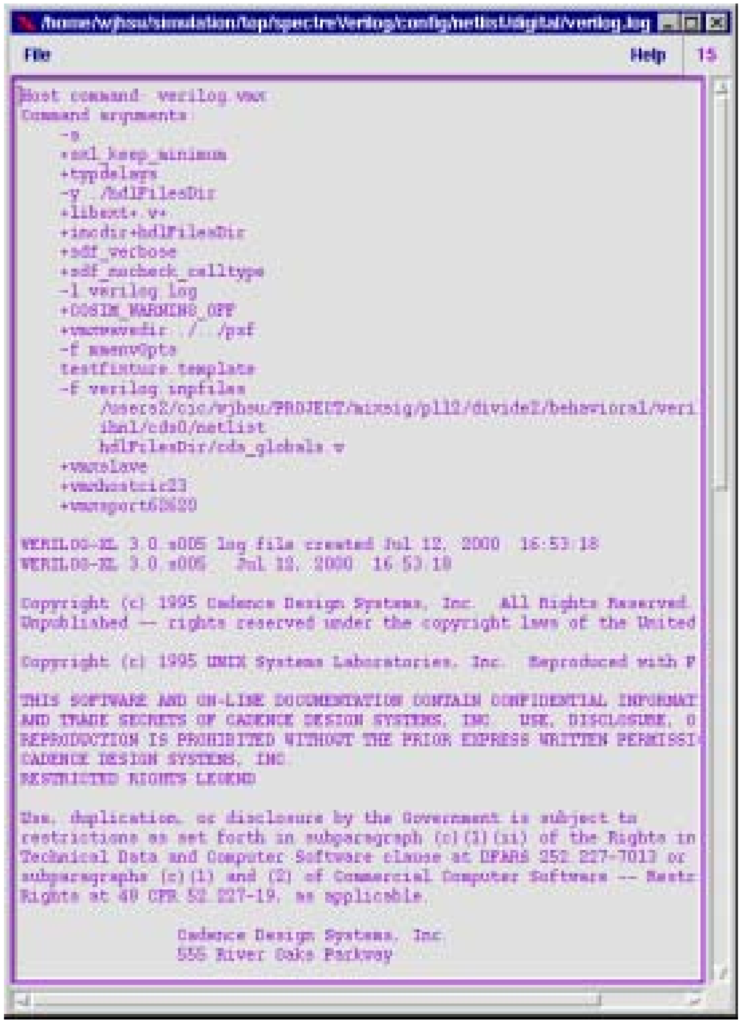
>

start simulation

set initial condition

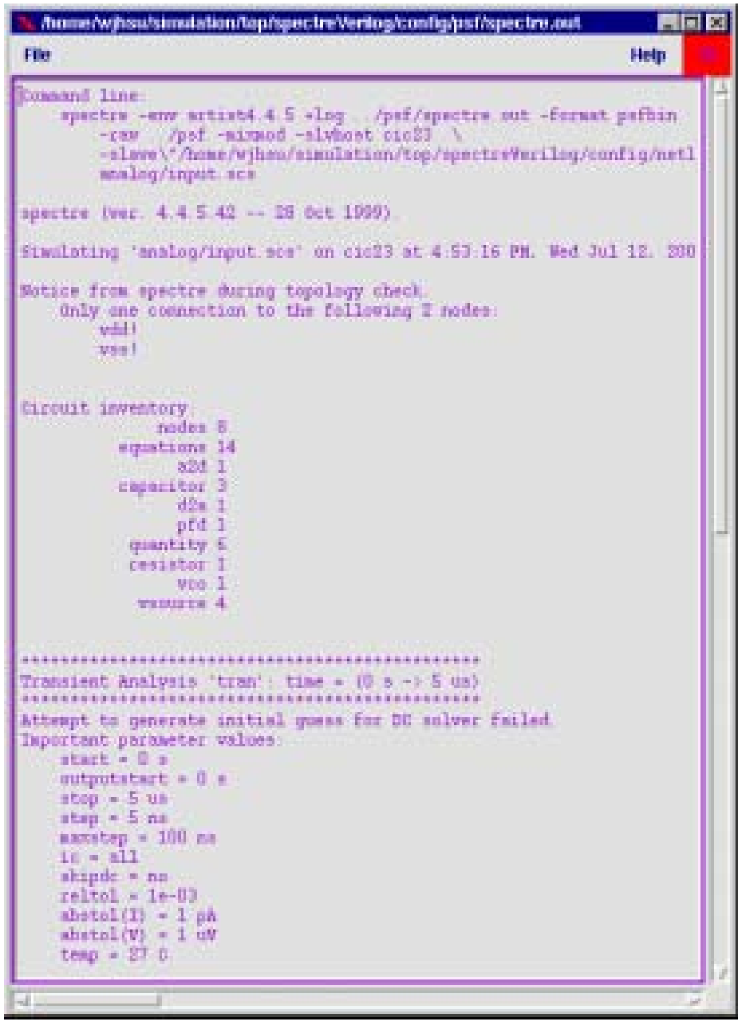
Run Log Files

Message for digital simulator


/home/wjhsu/simulation/top/spectreVerilog/config/netlist/digital/verilog.log
File
Help
15

```
Host command: verilog.vmx
Command arguments:
    -s
    +nxl_keep_minimum
    +typdelays
    -y ../hdlFilesDir
    +libext+.v+
    +incdir+hdlFilesDir
    +sdf_verbose
    +sdf_nocheck_celltype
    -l verilog.log
    +COSIM_WARNING_OFF
    +vmxwavedir ../../psf
    -f mmenvOpts
    testfixture.template
    -f verilog.inpfiles
        /users2/cic/wjhsu/PROJECT/mixsig/pll2/divide2/behavioral/veri
        ihnl/cds0/netlist
        hdlFilesDir/cds_globals.v
    +vmxslave
    +vmxhostcic23
    +vmxport62620

VERILOG-XL 3.0.s005 log file created Jul 12, 2000  16:53:18
VERILOG-XL 3.0.s005   Jul 12, 2000  16:53:18

Copyright (c) 1995 Cadence Design Systems, Inc.  All Rights Reserved.
Unpublished -- rights reserved under the copyright laws of the United

Copyright (c) 1995 UNIX Systems Laboratories, Inc.  Reproduced with P

THIS SOFTWARE AND ON-LINE DOCUMENTATION CONTAIN CONFIDENTIAL INFORMAT
AND TRADE SECRETS OF CADENCE DESIGN SYSTEMS, INC.  USE, DISCLOSURE, O
REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR EXPRESS WRITTEN PERMISSI
CADENCE DESIGN SYSTEMS, INC.
RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to
restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in
Technical Data and Computer Software clause at DFARS 252.227-7013 or
subparagraphs (c)(1) and (2) of Commercial Computer Software -- Restr
Rights at 48 CFR 52.227-19, as applicable.

                Cadence Design Systems, Inc.
                555 River Oaks Parkway
```

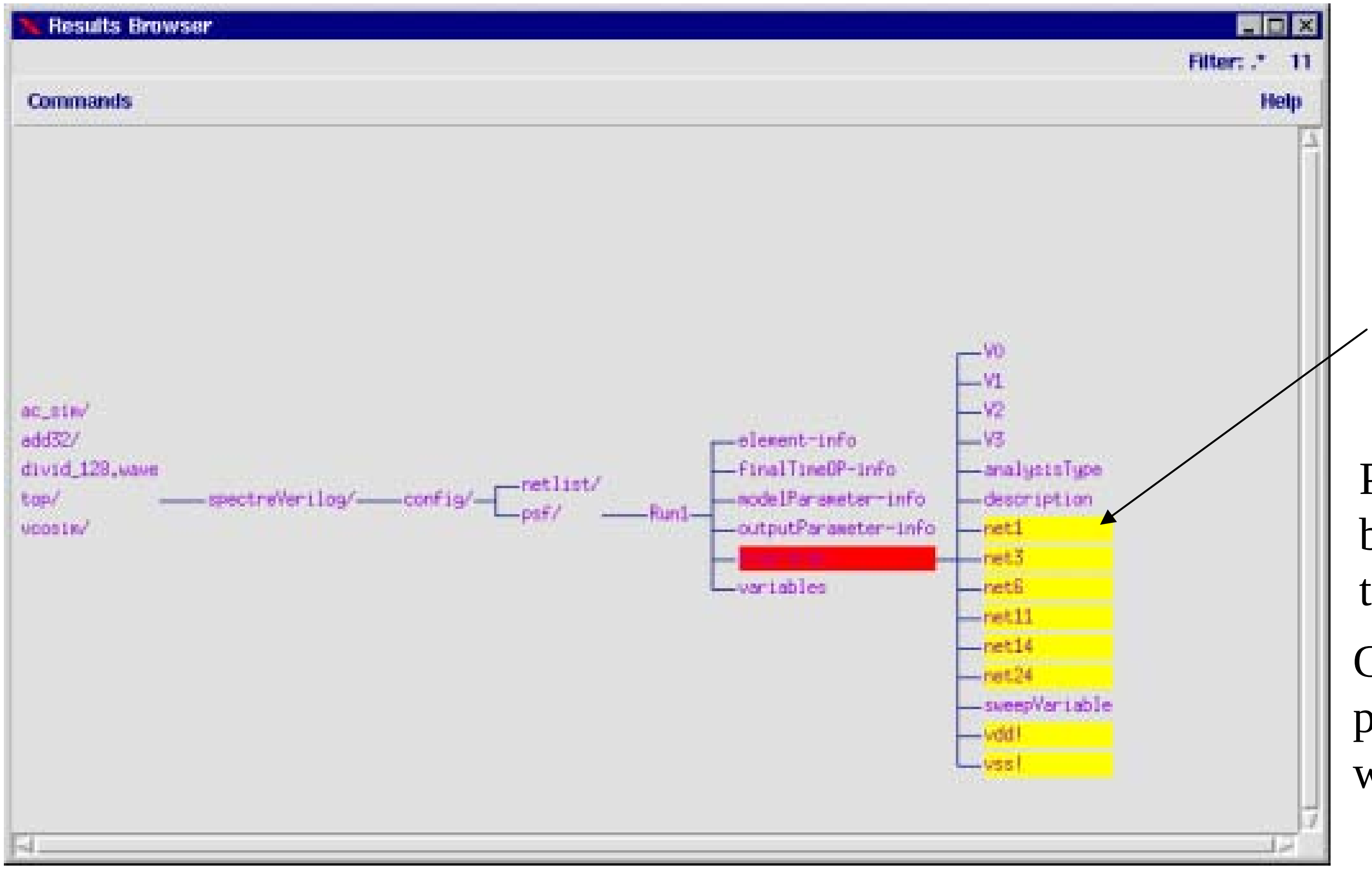
Message for analog simulator


/home/wjhsu/simulation/top/spectreVerilog/config/psf/spectre.out
File
Help

```
Command line:
    spectre -env artist4.4.5 +log ../psf/spectre.out -format psfbin
        -raw ../psf -mixmod -slvhost cic23  \
        -slave\"/home/wjhsu/simulation/top/spectreVerilog/config/netl
        analog/input.scs

spectre (ver. 4.4.5.42 -- 28 Oct 1999)

Simulating 'analog/input.scs' on cic23 at 4:53:16 PM, Wed Jul 12, 200

Notice from spectre during topology check.
    Only one connection to the following 2 nodes:
        vdd!
        vss!


Circuit inventory:
              nodes 8
          equations 14
                a2d 1
          capacitor 3
                d2a 1
                pfd 1
           quantity 6
           resistor 1
                vco 1
            vsource 4


***********************************************
Transient Analysis 'tran': time = (0 s -> 5 us)
***********************************************
Attempt to generate initial guess for DC solver failed.
Important parameter values:
    start = 0 s
    outputstart = 0 s
    stop = 5 us
    step = 5 ns
    maxstep = 100 ns
    ic = all
    skipdc = no
    reltol = 1e-03
    abstol(I) = 1 pA
    abstol(V) = 1 uV
    temp = 27 C
```

Result Browser

After several simulation jobs, you can choose any of the simulation results for waveform window



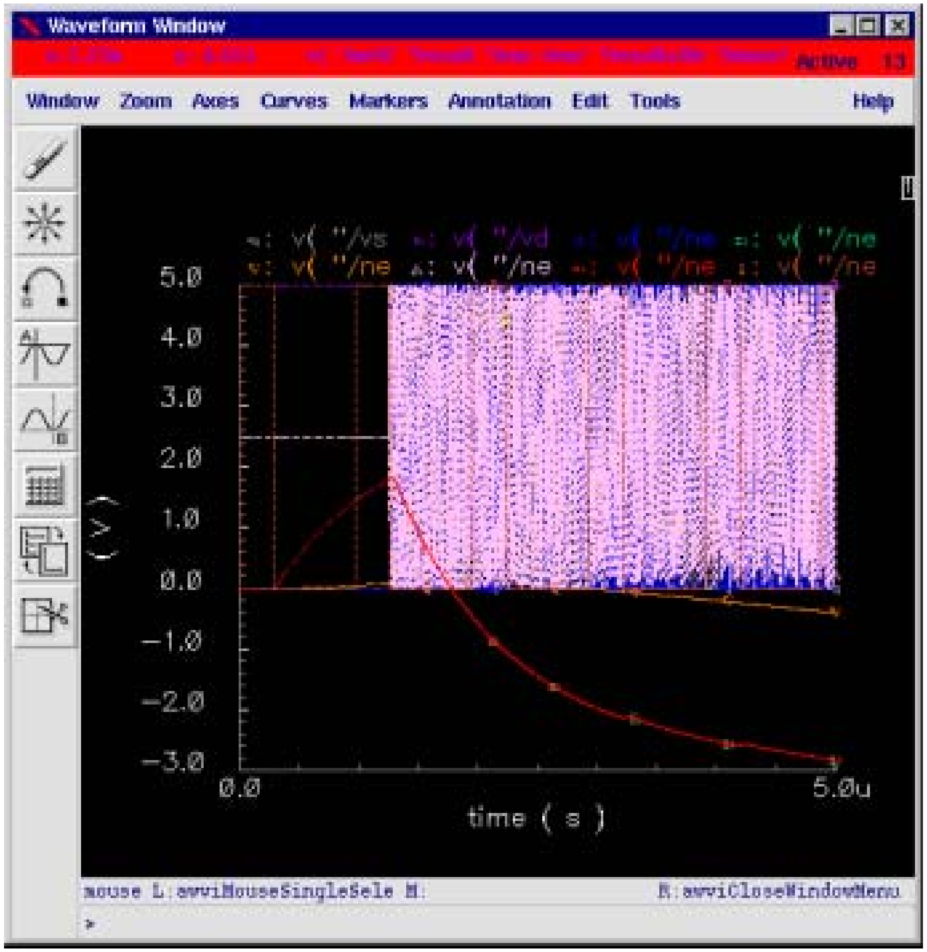
Saved node voltages

Press middle button to bring the menu

Choose **plot** to plot the waveform

Waveform Window

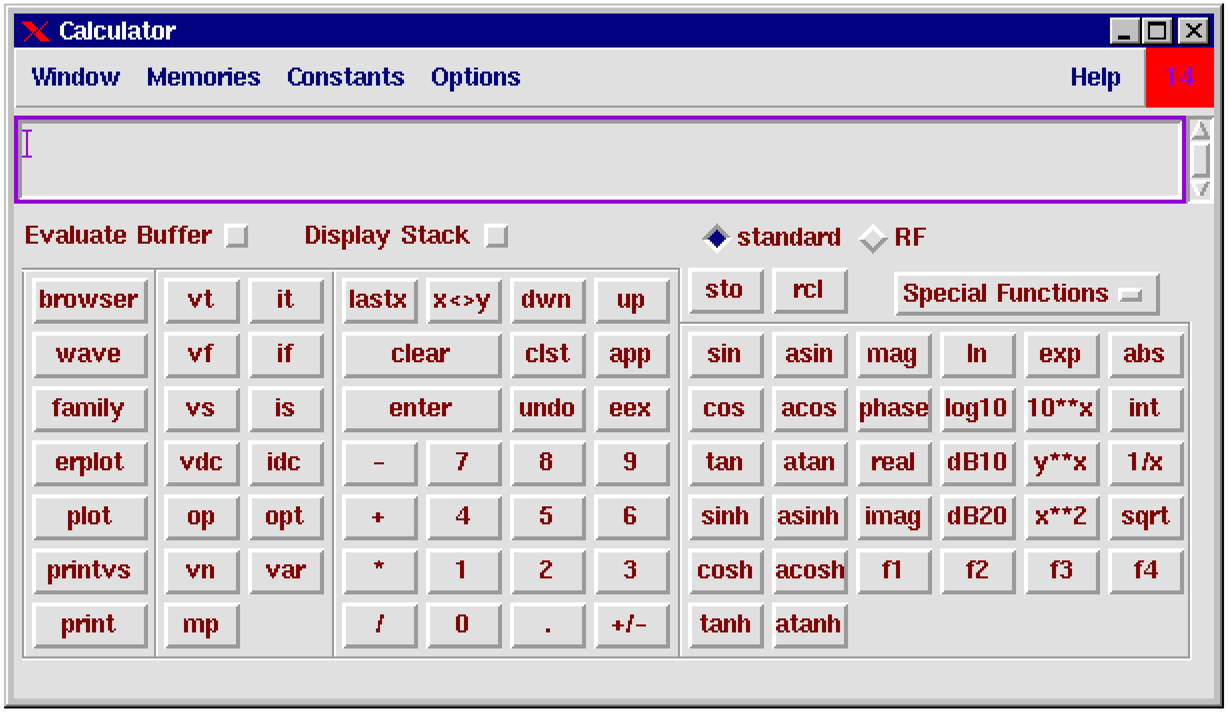
- After Simulation, the selected waveform will be shown in the window.
- The waveform can be digital waveform or analog waveform.
- The previously simulated data can also be shown through result browser.



Waveform Calculator

If further calculation is needed for obtaining other simulated parameters, the calculator can be used.

The calculator is invoked with **Tools → Calculator** in analog artist window

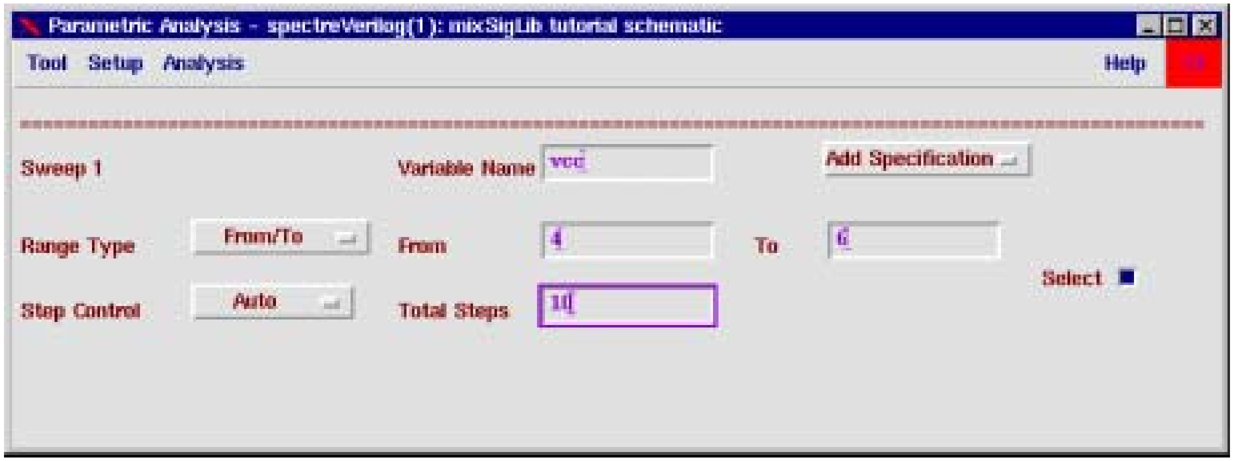


Evaluate Buffer □ Display Stack □ ◆ standard ◇ RF

Parametric Sweep

Invoke by Tool → Parametric Analysis

Nested simulation can also be used



Parametric Analysis - spectreVerilog(1): mixSigLib tutorial schematic

Sweep 1

Variable Name: vcc

Range Type: From/To

From: 4

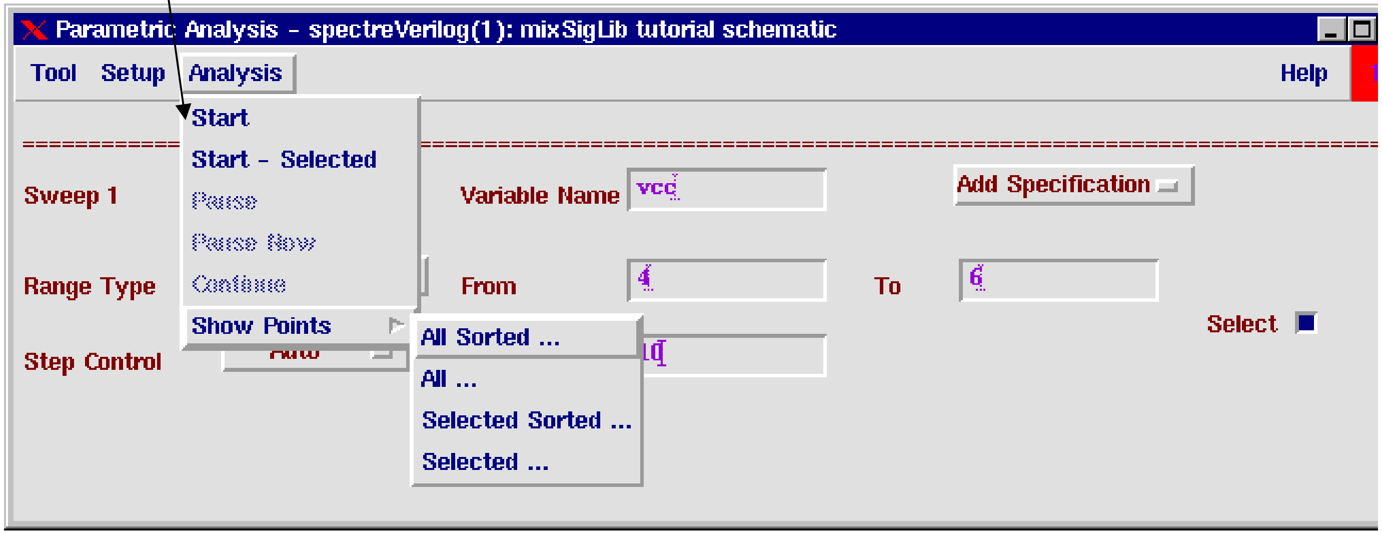
To: 6

Step Control: Auto

Total Steps: 10

Select ■

Start the sweep simulation



Parametric Analysis - spectreVerilog(1): mixSigLib tutorial schematic

Sweep 1

Variable Name: vcc

Range Type: From/To

From: 4

To: 6

Step Control: Auto

Total Steps: 10

Select ■

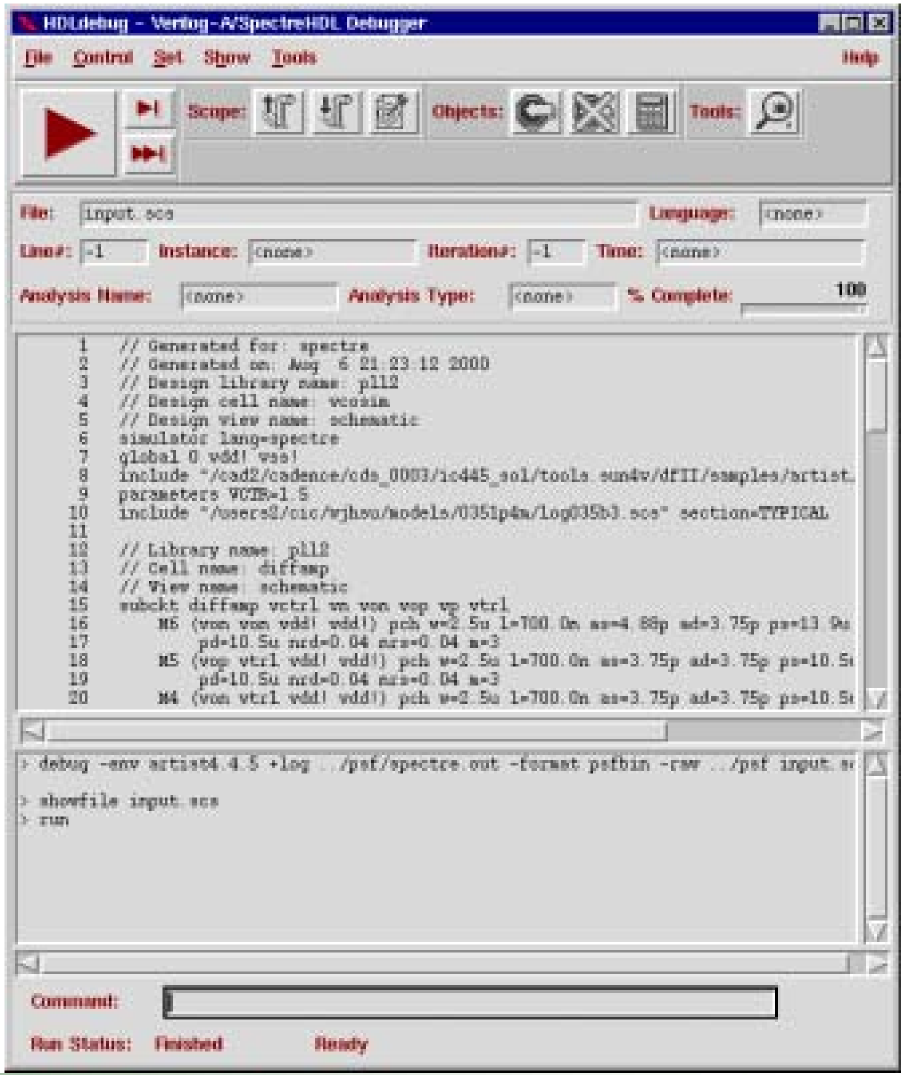
AHDL Debugger

For the HDL design, the Verilog-A is capable of interactive debug to help check the simulation process.

Simulation

- Netlist and Run
- Run
- Stop
- Options
- Netlist
- Output Log…
- Convergence Aids
- → Netlist and Debug AHDL
- or → Debug AHDL

To invoke the debugger


HDLdebug - Verilog-A/SpectreHDL Debugger
File Control Set Show Tools Help
Scope:
Objects:
Tools:

File: input.scs Language: <none>

Line#: -1 Instance: <none> Iteration#: -1 Time: <none>

Analysis Name: <none> Analysis Type: <none> % Complete: 100

```
// Generated for: spectre
// Generated on: Aug  6 21:23:12 2000
// Design library name: pll2
// Design cell name: vcosim
// Design view name: schematic
simulator lang=spectre
global 0 vdd! vss!
include "/cad2/cadence/cds_0003/ic445_sol/tools.sun4v/dfII/samples/artist.
parameters VCTB=1.5
include "/users2/cic/wjhsu/models/0351p4m/log035b3.scs" section=TYPICAL

// Library name: pll2
// Cell name: diffamp
// View name: schematic
subckt diffamp vctrl vn von vop vp vtrl
    M6 (von von vdd! vdd!) pch w=2.5u l=700.0n as=4.88p ad=3.75p ps=13.9u
        pd=10.5u nrd=0.04 nrs=0.04 m=3
    M5 (vop vtrl vdd! vdd!) pch w=2.5u l=700.0n as=3.75p ad=3.75p ps=10.5u
        pd=10.5u nrd=0.04 nrs=0.04 m=3
    M4 (von vtrl vdd! vdd!) pch w=2.5u l=700.0n as=3.75p ad=3.75p ps=10.5u
```

```
> debug -env artist4.4.5 +log ../psf/spectre.out -format psfbin -raw ../psf input.s

> showfile input.scs
> run
```

Command:

Run Status: Finished Ready

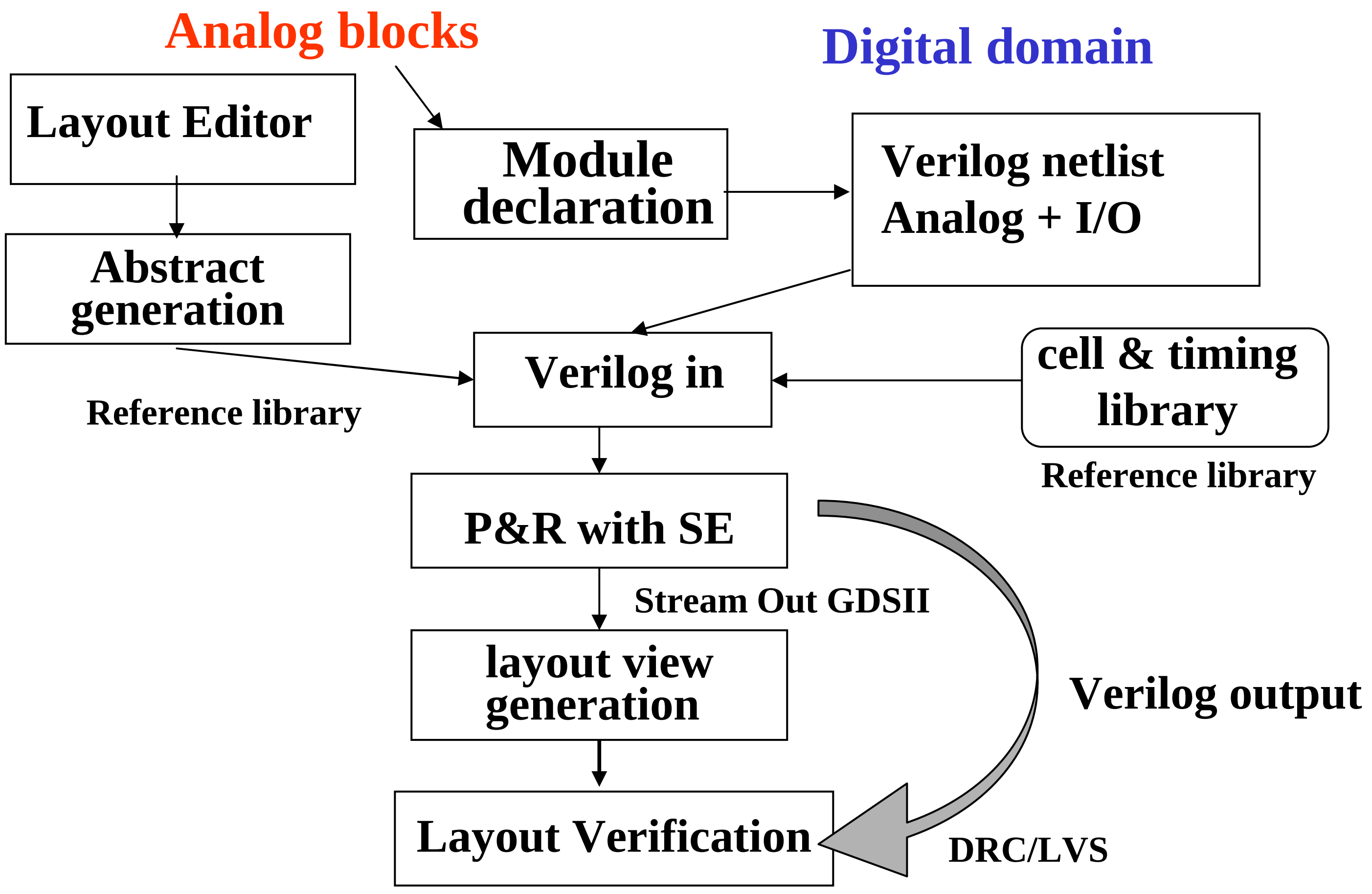
Feature of HDL debug

- Set breakpoints in modules
- Step line-by-line through code, step over, or step into functions and force a return from a function
- Run, stop, resume or reset a simulation
- View the value of variables, or display the changing values of variables as simulation progresses
- probe node/branch voltages and currents
- Move the scope up and down the function call stack, inspecting arguments and variables in user-defined functions.

Layout Integration and Verification for Mixed-Signal Design

Note : Most of the layout design issues follow the cell-based design approach, please refer to the *Cell-Based Physical Design and Verification* training manual and CIC PDS document *PDS-030612-01-000.pdf*

Layout Integration Flow



Prepare Data for SE

- ***LEF file***
- Verilog file
- CTLF/TLF file

Prepare Data for SE : LEF File

- Definition:

  LEF (Library Exchange Format) – pin and boundary information

- Place and Route tool needs pin and boundary information for block routing.
- The manual layout contains only geometry information, need to define pin location and information for P&R tool.

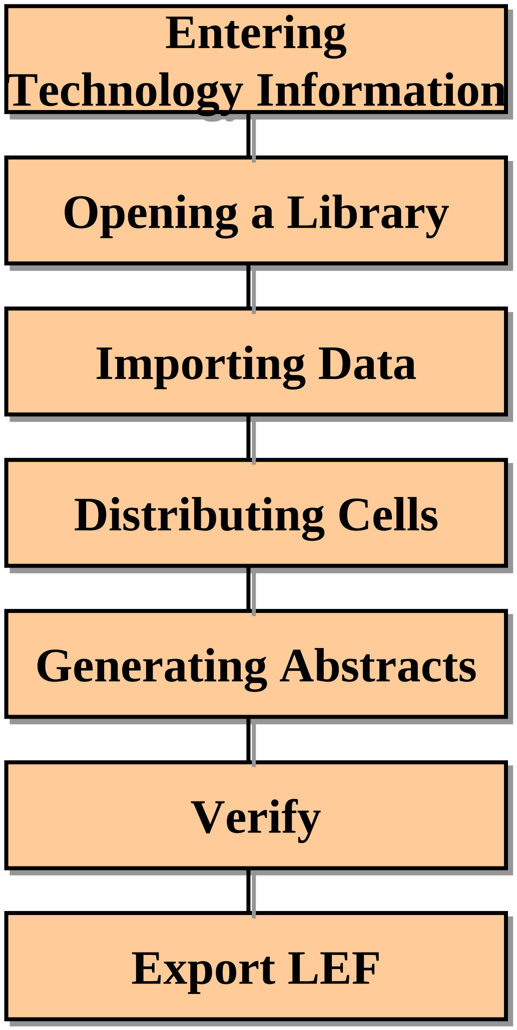
LEF File of TSMC .35um 2P4M

- Digital core & IO:
  - Standard cell: ***tcb773p_4lm-cic.lef***
  - IO pad: ***tpz773pn_4lm-cic.lef***
- Analog IO:
  - IO pad: ***tpz773pn_analog_4lm.lef***
- Custom analog block:
  - Generate the LEF file by ***Cadence Abstract Generator***

Cadence Abstract Generator

- Cadence Abstract Generator is a library modeling tool for creating layout abstracts from detailed layout information for standard cells, macro blocks, and IO cells.
- The abstracts generated are based on physical (layout) and logical data, process technology information, and specific cell-modeling requirements.
- After abstract generation, a LEF file will be generated for Silicon Ensemble Verilog in.

Cadence Abstract Generator Flow



Entering Technology Information

Supply Information about process technology from an existing tech.dpux file, by mapping GDSII files, or by importing LEF or DFII.

Opening a Library

Set the location where your cell views are stored.

Importing Data

Import layout (Stream GDSII or DEF) and/or logical data (Verilog or TLF).

Distributing Cells

Partition library cells into mutually exclusive sets for processing.

Generating Abstracts

Run the three abstract generation steps – Pins, Extract, and Abstract.

Verify

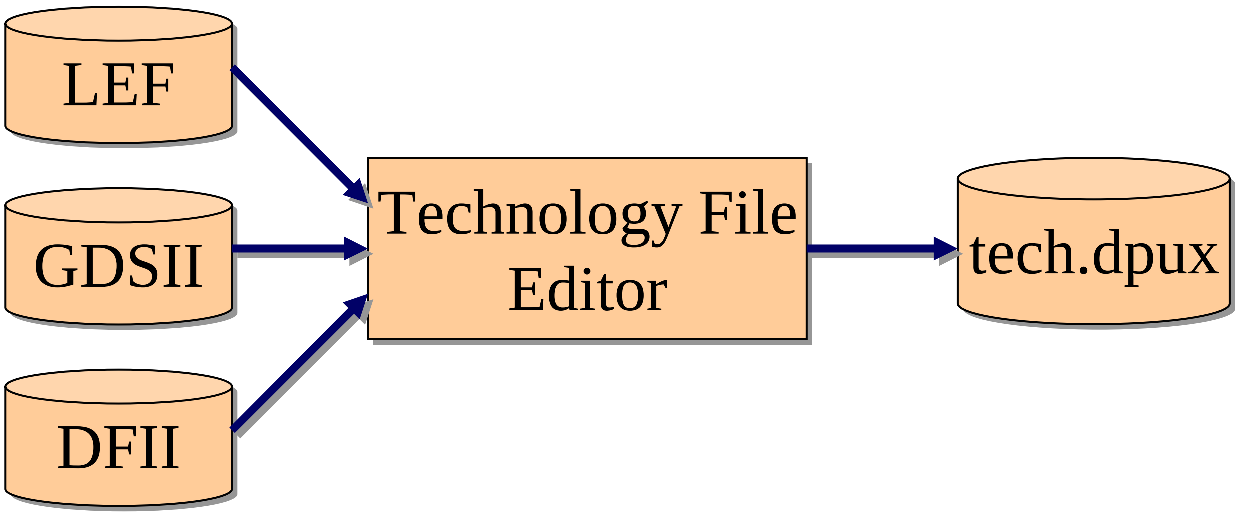
Run checks to detect any problems in the abstracts generated.

Export LEF

Generate LEF descriptions of the abstracts generated.

Entering Technology Information

- Technology information provides details of the process technology to be used during IC fabrication, including names of layers, colors, and fill patterns; GDSII layer mapping data; and design rules for different layers and vias.

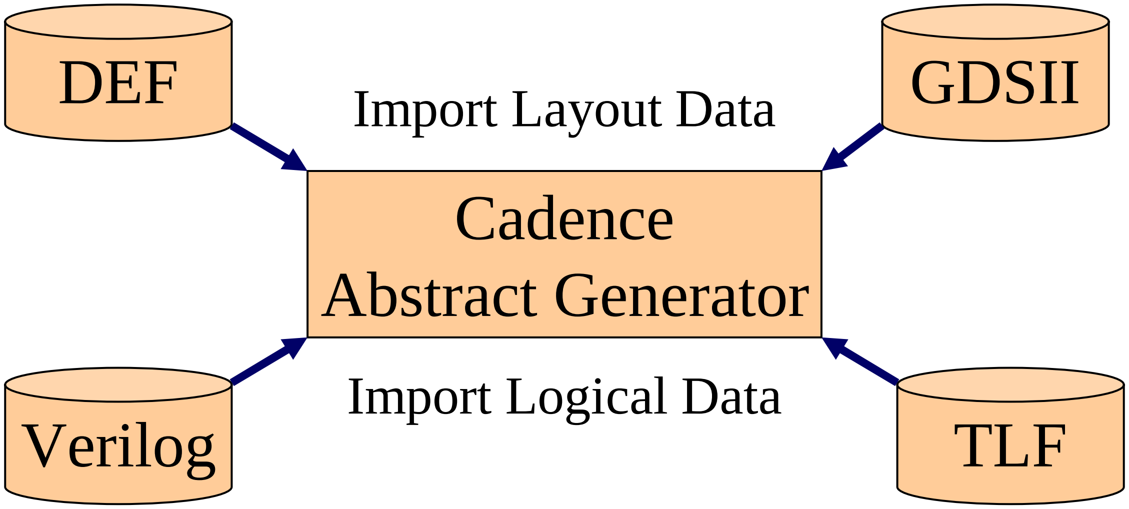


Opening a Library

- Included in the technology information are the names and paths of the library or libraries you want to process. The Abstract Generator reads in this information and lists the libraries in the *Library* form.

Importing Data

- After you specify technology information and set the library, you have to import information on the cells for which you want to create abstracts.



Distributing Cells

- Central to the Cadence Abstract Generator is the concept of “bins”, which provide a simple means of distributing the cells in a library into a number of mutually exclusive sets.
- Four main types of cells can be processed: Core, IO, Corner, and ***Block***. The other *Ignore* bin can be used to store cells that you do not want to process.

Generating Abstracts

- *Pins:*
  - The Abstract Generator creates a place-and-route boundary for the cell and starting pin shapes for each of the nets to be extracted.
- *Extract:*
  - The Abstract Generator derives which shapes are connected to which nets by tracing the connectivity from the *pin* purpose shapes created during the Pins step.

Generating Abstracts

- *Abstract:*
  - The Abstract Generator adjusts the pin shapes created during the extract step to create the final shapes required by the place-and-route tools. It then fractures these pin shapes into rectangles.
  - Next, the Abstract Generator applies a layer blockage model selected by the user to create the final blockage geometry in the abstract. The blockage geometry is then fractured into rectangles.
  - It then removes from the abstract all layers other than those with purpose *pin*, *blockage*, or *boundary*, and deletes the instance hierarchy. At this stage all the required geometry is at the top level of the abstract.
  - The Abstract Generator then annotates grid lines onto the abstract view and calculates grid pitches if these are not already present. It adjusts the placeand-route boundary to be a multiple of the grid pitches.

Verify

- *Verify*:
  - During the verify step, terminals are compared for any differences that might exist between logical and abstract views. Pin and geometry information on manufacturing grids is checked and each abstract is tested within the target place-and-route system.

Exporting

- If you are using Silicon Ensemble to place-and-route a design that references cells contained in your library, you will have to generate LEF descriptions of your abstracts. *Export – LEF* provides this function, translating the abstracts into library exchange format, which can be used as input to Silicon Ensemble.
- Modify analog block LEF file, ***only keep Macro data***. Remove the Technology and Row data form analog block LEF file.

Prepare Data for SE

- LEF file
- ***Verilog file***
- CTLF/TLF file

Prepare Data for SE : Verilog File

- In order for P&R tools to route the connect between analog and digital blocks, the analog block must be in verilog netlist and connected.
- If you are designing a chip, you have to add io pads and power pads into the netlist before you import it.
- You can import either a verilog or a DEF as your design netlist.
- A standard cell verilog model is needed as verilog reference.

Verilog File of TSMC .35um 2P4M

- Digital core & IO:
  - Standard cell: ***tcb773p.v***
  - IO pad: ***tpz773pn-cic.v***, ***filler.v***
- Analog IO:
  - IO pad: ***tpz773pn_analog-cic.v***
- Custom design:
  - Digital & Analog module, including IO pads

Module Declaration

- Since there might not be any Verilog statement that can model the analog block, the analog block contains only I/O port information.
- Must be enabled **Don’t Touch** when synthesis.
  - Using Attributes → Optimization Directives → Design

```
module A_module_1(J, T, Y) ;
  input [7:0] J, T ;
  output Y;
endmodule
```

```
module core_top(clk, In, Y)
  input clk ;
  input [5:0] In ;
  Output Y ;

  latcha  La1 ( clk, In, J ) ;
  A_module_1 IM1 ( J, T, Y) ;
endmodule
```

Top Level Verilog Netlist Integration

- The final chip must contain I/O pads, these I/O information were also provided by Verilog netlist.

```
module CHIP ( Tclk, TBin, TY );
  input Tclk;
  input [5:0] TBin;
  output TY;

  wire Wclk;
  wire [5:0] WBin;
  wire WY;

//OUTPUT IO PAD C
  PDIANA2P oY(TY,WY);

//INPUT IO PAD C

  PDIZ iCLK(.PAD(Tclk),.C(Wclk));

  PDIZ iBIN0(.PAD(TBin[0]),.C(WBin[0]));
  PDIZ iBIN1(.PAD(TBin[1]),.C(WBin[1]));
  PDIZ iBIN2(.PAD(TBin[2]),.C(WBin[2]));
  PDIZ iBIN3(.PAD(TBin[3]),.C(WBin[3]));
  PDIZ iBIN4(.PAD(TBin[4]),.C(WBin[4]));
  PDIZ iBIN5(.PAD(TBin[5]),.C(WBin[5]));
```

```
//POWER IO
                                   I/O powers

  PVSS2Z  VSS_0(.IOVSS(IOVSS));
  PVDD2Z VDD_0(.IOVDD(IOVDD));
  PVSS3P  VSS_1(.TAVSS(TAVSS));
  PVDD3P VDD_1(.TAVDD(TAVDD));
                                   internal power

  PVSS1Z  INT_VSS_0(.VDD(VDD));
  PVDD1Z INT_VDD_0(.(VSS(VSS));
  PCORNERZ corner1(); ←——— corner cells
  PCORNERZ corner2();
  PCORNERZ corner3();
  PCORNERZ corner4();


 core_top TOP ( Wclk, WBin[5:0], WY );

endmodule
                                 core cell
```

Prepare Data for SE

- LEF file
- Verilog file
- ***CTLF/TLF file***

Prepare Data for SE : CTLF/TLF File

- Definition:

  CTLF (Compiled Timing Library Format)

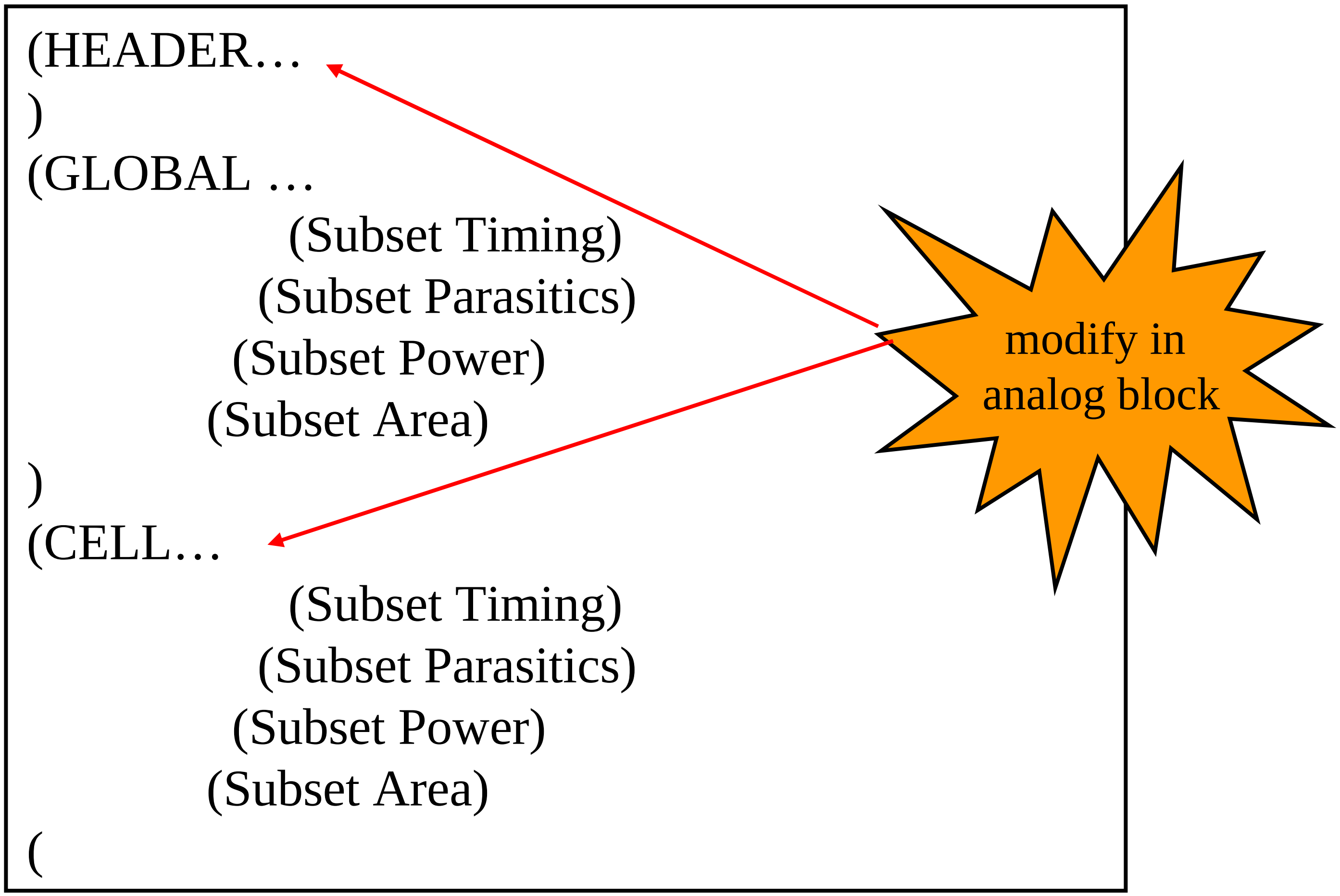
- You need to import compiled timing library (CTLF) by reading in GCF (general constraint format) file for exporting Verilog file.

GCF File

environment.gcf

```
(gcf
 (header
 (version "1.3")
 (TIME_SCALE 1.0E-9)
 (CAP_SCALE 1.0E-12)
)
 (globals
   (globals_subset environment
     (process 1.0 1.0)
     (voltage 3.3 3.3)
     (temperature 0 100)
     (extension "CTLF_FILES"
       ('tlf/tcb773pwc.ctlf'
    'tlf/tpz773pnwc-cic.ctlf'
    'tlf/tpz773pn_analogwc-cic.ctlf'
    'tlf/IMatrix8x8.ctlf'
        )
     )
     (operating_conditions "typical" 1.0 3.3 25.0)
   )
  )
)
```

TLF Format



```
(HEADER…
)
(GLOBAL …
                    (Subset Timing)
                  (Subset Parasitics)
                (Subset Power)
              (Subset Area)
)
(CELL…
                    (Subset Timing)
                  (Subset Parasitics)
                (Subset Power)
              (Subset Area)
(
```

modify in analog block

Example of TLF File

```
…
Header(
    Library("IMatrix8x8")
    Date(" ")
…
Cell(IMatrix8x8
    Celltype(Seq)
    Model(ioDelay
…
 Pin(J[6:0]
        Pintype(Data)
        Pindir(Input)
…
Pin(T[6:0]
        Pintype(Data)
        Pindir(Input)
…
```

```
Pin(Y
        Pintype(Data)
        Pindir(Output)
…
Path(J[6:0] *> Y  01 01 Delay(ioDelayRiseModel0)
         Slew(SlopeRiseModel0))
    Path(J[6:0] *> Y  10 01 Delay(ioDelayRiseModel0)
         Slew(SlopeRiseModel0))
    Path(J[6:0] *> Y  01 10 Delay(ioDelayRiseModel0)
         Slew(SlopeRiseModel0))
    Path(J[6:0] *> Y  10 10 Delay(ioDelayRiseModel0)
         Slew(SlopeRiseModel0))

    Path(T[6:0] *> Y  01 01 Delay(ioDelayRiseModel0)
         Slew(SlopeRiseModel0))
    Path(T[6:0] *> Y  10 01 Delay(ioDelayRiseModel0)
         Slew(SlopeRiseModel0))
    Path(T[6:0] *> Y  01 10 Delay(ioDelayRiseModel0)
         Slew(SlopeRiseModel0))
    Path(T[6:0] *> Y  10 10 Delay(ioDelayRiseModel0)
         Slew(SlopeRiseModel0))
)
```

Compiled TLF File

- SE only reads in the compiled timing library format.
- Using ***tlfEncrypt*** command to generate the CTLF file.
  - /usr/cadence/DSMSE/cur/tools/tlfUtil/bin/tlfEncrypt
  - Usage: tlfEncrypt <input_file> <out_file>

CTLF File of TSMC .35um 2P4M

- Digital core & IO:
  - Standard cell: ***tcb773pwc.ctlf***
  - IO pad: ***tpz773pnwc-cic.ctlf***
- Analog IO:
  - IO pad: ***tpz773pn_analogwc-cic.ctlf***
- Custom analog block:
  - Create TLF file and use ***tlfEncrypt*** command to generate the CTLF file

Layout Integration

- After preparing the data for SE, standard Place & Route flow with hard macro can be used for mixed-signal circuit layout generation.
- The major differences are the I/O pads and power/ground route.
- 0.35um cell library used TSMC I/O pads for digital pads and analog pads.

Place IO Constraints

Create ***placeIO.ioc*** file for SE Place IO.

IO_pad constraint

```
LEFT(
    |ipad_I1        East;
)
RIGHT(
    |opad_o1        West;
)
TOP(
    |DCVSS        South;
)
BOTTOM(
    |DCVDD        North;
)
```

No_IO_pad constraint

```
LEFT(
    (IOPIN I1)
)
RIGHT(
    (IOPIN O1)
)
TOP(
    (IOPIN I2)
)
BOTTOM(
    (IOPIN I3)
)
```

SE Startup

- Before start SE, create a directory named ***dbs*** to store saved data. (Please use SE ***v5.4*** above)
- Copy **se.ini** file, CIC provided to working directory.
- In Unix command line, enter

```
seultra -m=200&
```

SE start with limit 200 MB virtual memory.

- The initial environment is read form **se.ini** file.
- You can set selectability(SI) and visibility(VI) form object selection windows.

Import LEF File

- Before creating a design database, you need a library database that contain technology rule and information of cells that used in your design.
  - Import LEF file: ***tcb773p_4lm-cic.lef***, ***tpz773pn_4lm-cic.lef***, and ***tpz773pn_analog_4lm.lef***
- Add analog block information by the same way for SE.
- Save your design at some step is always a good ideal.

Import Timing Library

- After APR, it needs to export Verilog for LVS. In addition, to export Verilog needs the timing library. Use import Timing commands by adding the ***environment.gcf*** provided by CIC to the timing library database.
- Environment.gcf file defines the path of timing library.
  - CIC provided ***tcb773p-cic.ctlf***, ***tpz773pn-cic.ctlf***, and ***tpz773pn_analog-cic.ctlf*** file for SE to import timing library.
  - The timing library of analog block should be also added the same way for SE.

Import Verilog File

- CIC provided ***filler.v***, ***tcb773p.v***, ***tpz773pn-cic.v***, and ***tpz773pn_analog-cic.v*** file for SE to import Verilog as the library reference.
- To import the verilog file of analog block by the same way for SE.
- The verilog reference library should be imported first, then importing the chip verilog files.

Initiate Floorplan

- Set the two environment variables to ensure io row and corner row contact tightly.

  Set var plan.iorow.snapgrid.x 1

  Set var plan.iorow.snapgrid.y 1

- Enable ***Flip Every Other Row*** and ***Abut Rows*** with Row Spacing “0” on Core Area Parameters.
- Let Block Halo Per Side the same as IO to Core Distance.
  - Analog Block includes power rails, SE just can offer one power rails for digital block.
- SE automatically calculate the core area form given value.
- In CIC 0.35 process, METAL1 and METAL3 are defined as horizontal routing layers, METAL2 and METAL4 are defined as vertical routing layers.

Place Analog Block

- Select the block and move it to the desired location.
- Place blocks with Optimize routablility by setting Span On.
- Place blocks along the edges of core area, preferably in a corner of the core.
- Use Floorplans → Update Core Rows after all blocks placed.
- Set Global Block Hale in Update Core Rows windows to reserve spacing for block power ring.

Plan Power

- Use Route → Plan Power, delete power path that analog block does not routed.
- Modify the following variables in PP Add Rings form.
  - Nets name: “VDD VSS”
  - Ring Layer:

    METAL3 for Horizontal

    METAL2 for Vertical
  - Core Ring Width the same as Block Ring Width
- Modify the variables in PP Add Stripes form, if you need.

Add PRDIODE

- To support a separate the analog and digital power scheme
  - Using power cut cells(***PRDIODE***), if you want to have clear power for analog block or your design has two or above kinds of analog power provider PVDDXPX and PVSSXPX.
  - Use Place → Filler Cells → Add Cells
    - Model: PRDIODE
    - Prefix: PRDIODE
    - Area: Click and drag across the area you want with left mouse button.
- The add prdiode pads can NOT be remove.
- Do this step MUST be before then Add IO Filler step.

Add IO Filler

- Connect io pad power bus by insert IO filler.
- Add form wider filler to narrower filler.
  - Use Place → Filler Cells → Add Cells
    - Model:

      PFEED20Z, PFEED10Z, PFEED8Z, PFEED5Z, PFEED4Z, PFEED2Z, PFEED1Z
    - Prefix:

      PFEED20Z, PFEED10Z, PFEED8Z, PFEED5Z, PFEED4Z, PFEED2Z, PFEED1Z
- The add IO fillers can NOT be remove.
- Do this step MUST be before then WRoute step.

Placement

- ***Place -> Cells***
- Qplace include three placement phases
  - Global placement phase partitions the netlist hierarchically and initially places the cells
  - Detailed placement phase swaps individual cell location
  - Annealing refinement phase snaps cells to rows and remove overlaps

Connect Rings

- Stripe
  - Connect stripes to the closet power ring
- Block
  - Connect block pins to the closet power ring
- IO PAD
  - Connect power pad to power ring
- IO Ring
  - Run SROUTE FOLLOWPIN on IO ROWs
- Follow Pins
  - Run SROUTE FOLLOWPIN on CORE ROWs

Use Ultra Router

- The ultra router consists of several phases.
  - Global routing
  - Final routing
  - Search and repair
  - Final clear-up

Verification for Mixed-Signal Design

Note : Mentor Calibre is used in TSMC 0.35um 2P4M for verification flow (DRC, LVS, LPE). Please refer to CIC PDS document, *PDS-030616-00-000.pdf*

Prepare for Layout Verification

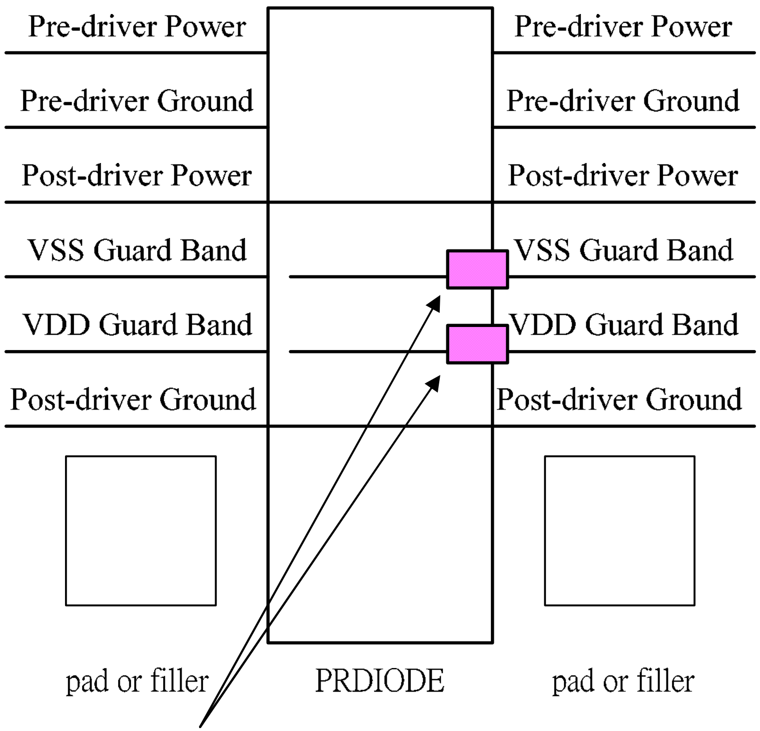
- Using Calibre DRC and LVS for layout verification.
- In SE, you need to export Verilog and GDSII file:
  - using File → Export → Verilog… to export Verilog file.
  - using File → Export → GDSII… to export GDSII file.
    - Choose the ***gds2.map*** file to add text
    - Set suitable value of ***OUTPUT.ORIGIN.X*** and ***OUTPUT.ORIGIN.Y*** to avoid the DRC error

Read into DFII Library

- After exporting GDSII file from SE, this GDSII file need to be read into DFII library first. Then, stream out GDSII file from DFII library for DRC, LVS, and tape out.
- In CIW, using ***File→ Import→ Stream*** to import the GDSII file generated by SE.
  - In options, turn on ***Case Sensitivity ”preserve”*** and ***Retain Reference Library (No Merge)***
  - Make sure the reference library can be found in Library Manager (modify the cds.lib). It’s better that there is only the reference library in Library Manager.

Modify Layout

- As the layout of PRDIODE reveals, the guard bands within the PRDIODE are not aligned with the left and right side of the cell boundary.
- For VDD band connect side:
  - The highest VDD
  - The most IO pad
  - The analog power
- For VSS band connect side:
  - The lowest VSS
  - The most IO pad
  - The analog power



- Connect layer ONLY using **METAL4** drawing

Export Layout

- In Virtuoso, delete the original power and IO routing of analog block. Then, route the suitable wide line for power and IO pin of analog block.
- In CIW, using ***File → Export → GDSII***
  - In options, turn on ***Case Sensitivity "preserve"***, but turn off ***Retain Reference Library (No Merge)***

Calibre DRC Flow

- Edit the Calibre DRC runset file (***tsmc35DRC.cal***):
  - Layout Path
  - Layout Primary
  - Include ‘Calibre-drc-cur’, the Calibre-drc-cur is linked to the DRC command file “***CM35S5_4M.22b***”
- Run the batch mode of Calibre DRC:
  - ***calibre –drc –hier tsmc35DRC.cal***

Calibre LVS Flow

- Using "***v2lvs***" command of Calibre to translate the Verilog file from SE to the netlist for LVS.
  - Add the verilog module and subckt of analog block in tsmc35_lvs_ms.v and tsmc35_lvs_ms.spi, respectively.
  - ***v2lvs -v Verilog -l tsmc35_lvs_ms.v -o Output_netlist -s tsmc35_lvs_ms.spi -c cic -n***
- Edit the Calibre LVS runset file (tsmc35DRC.cal):
  - Layout Path & Primary
  - Source Path & Primary
  - Include 'Calibre-lvs-cur', the Calibre-lvs-cur is linked to the DRC command file "***cali035pMM5V_2P4M.lvs***"

Calibre LVS Flow (Cont.)

- Add analog block as ***LVS BOX*** in LVS command file (cali035pMM5V_2P4M.lvs)
  - ***LVS BOX Analog_Block_Name***
- Run the batch mode of Calibre LVS:
  - ***calibre –lvs –spice layout.spi –hier –auto tsmc35LVS.cal***

Post-Layout Simulation for Mixed-Signal Design

Note : Please refer to the *TimeMill/PowerMill/PathMill* training manual

Mixed Signal Parasitic Simulation

- Circuit in realistic contains parasitic elements which will affect system performance
- Post- layout simulation include parasitic elements into the complete design for verifying the overall design performance.
- Factors should be considered :

  What elements to be modeled ?

    - Interconnect loading capacitance
    - Interconnect wire resistance
    - Interconnect coupling capacitance
    - Power/Ground parasitic elements
    - Substrate resistance

  What flow to be used ?

Flows for Parasitic Simulation

- Complete elements extraction with layout extraction tools(such as Calibre)

  The extracted netlist is in SPICE netlist format , use spice tools or transistor level simulator(such as timemill, star-sim ...) for simulation.

- Separated extraction for digital and analog netlist

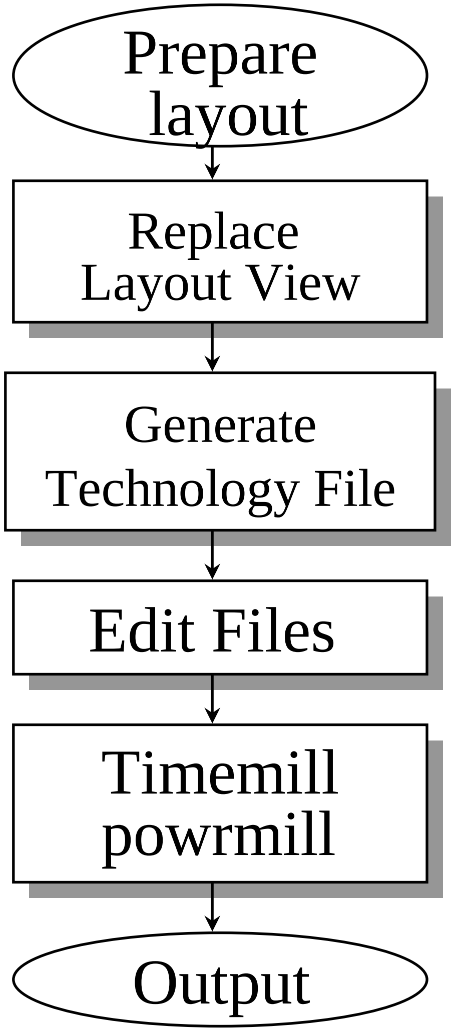
  Digital - Standard Delay Format(SDF)

  Analog-SPICE netlist

Transistor Level Post-Layout Simulation

- Due to the restriction of cell library, the complete chip simulation with Cadence MSPS flow might be troublesome.
- Use timemill for post-layout simulation at current stage.
- The tool is now installed in CIC, invoking with queue system.
- Refer to the timemill training manual for timemill usage flow.

Timemill Job Flow



Prepare layout

Replace Layout View

Generate Technology File

Edit Files

Timemill powrmill

Output

Ftp layout to CIC’s account

qepiclpe35a: replace cell layout view

gentech : generate technology file

vi/edit stimulus file, configuration commands

perform simulation

Timemill Input Stimulus File

- Create input stimulus file which specify type of input signals

  SIN voltage source

  - TimeMill syntax:

    (t=VSIN)(en=element_name)(so=n+)(dr=n-)(v=dc,pa,<freq,<td,<df,<pd>>>>);

  - SPICE syntax:

    Vname n+ n- dc SIN(dc pa <freq <td <df <pd>>>>)

  EXP voltage source

  - TimeMill syntax:

    (t=VEXP)(en=element_name)(so=n+)(dr=n-)(v=v1,v2,<td1,<tau1,<td2,<tau2>>>>);

  - SPICE syntax:

    Vname n+ n- EXP(v1 v2 <td1 <tau1 <td2 <tau2>>>>)

Reference

- Mixed-Mode Simulation and Analog Multilevel Simulation Resve Saleh, Shyh-Jye Jou, A. Richard Newton Kluwer Academic 1994
- Affirma Mixed-Signal Circuit Design Environment User Guide, Cadence
- Affirma Analog Artist Mixed-Signal Design Series, cadence
- Analog Modeling with Verilog-A,cadence
- Understanding Mixed-Signal Simulation, http://www.vhdl-ams.com/literature_link.htm
- VHDL-AMS Guide to Mixed-Signal Simulation,
  http://www.vhdl-ams.com/literature_link.htm