Hypermodel Library Automatic Generation

Ken G. Ruan

Synopsys Inc. 2025 nw Cornelius Pass Rd, Hillsboro, OR 97124
TEL: (503) 547-6399, Email: ken.ruan@synopsys.com

Abstract

In simulation of a mixed-signal IC, a hypermodel must be inserted if a digital port and an analog port are connected. In order to obtain reliable circuit behavior, Terminal characteristics of the hypermodel should be close enough to that of the digital circuit output.

This article describes the methodology that a hypermodel library can be generated automatically based on the information of a cell library of designer's selection. A program is developed based on this methodology.

Comparison of simulation results on characteristics of circuit level cell model and that of a selected hypermodel is also provided. The results show good fit and the generated hypermodel library can be used confidently in mixed-signal IC simulation.

1. Introduction

In IC design, a designer often works with a library of standard cells provided by a semiconductor vender. In this library, the circuit level model of each cell are included. In simulation of a mixed-signal IC, a hypermodel must be inserted if a digital port and an analog port are connected. In order to obtain circuit characteristics that are reasonably accurate, the terminal characteristics of the inserted hypermodel should be close enough to that of the particular output of the digital circuit. The existing approach to deal with this issue is that a number of pre-characterized hypermodel libraries based on manufacturers' specifications are provided and the user may select one that fits his/her design needs the best. Unfortunately, this approach is not suitable in IC design, because the cell libraries used in a design are proprietary and may change from time to time. It is not practical to provide hypermodel libraries by an EDA tool vender. The best way to address this issue is to create a tool that can generate a hypermodel library bast on user's cell library and model parameters.

This paper describes a methodology for hypermodel library generation that is based on the user's cell library information. Assume the circuit level model of each cell is collected in a source directory, the designer provides the information in a specification file that is in a straight forward format. The specification file is read by the tool and a test circuit for each cell will be created, accordingly. Simulation on each testbanch is performed and required characteristics are measured from the simulation results for each input and output ports of a cell. Hypermodel is then characterized to meet the measured characteristics for each port, individually. Finally, a hypermodel library is generated in the form that can be used by the netlister.

A program is developed based on this methodology.

Comparison of characteristics of circuit level cell model and that of the hypermodel is also provided. The results fit good with the measured characteristics and can be used confidently in mixed-signal IC simulation.

This paper is arranged as follows. In the next section, the specification file is described. Hypermodel generation methodology is described in Section 3. Results comparison is provided in Section 4 followed by the conclusion section.

2. Cell library and specification

A cell library is a library that contains the basic circuit elements used for a design. A cell can be as simple as a logic gate, or a functional block such as a counter, a register, an op amp, etc. Model parameters for each cell are predefined by the manufacturer that the characteristics of a cell are known. A designer, in general, works with a given cell library. Only elements that are available in that cell library may be used in circuit design.

Therefore, for any cell library, the following information is given:

1. For each cell, the circuit of the cell is known. In general, the vender provides a model for each cell;
2. Model arguments for each device such as mosfet, diode, bjt, etc. are defined based on manufacturing process;
3. A cell may be scalable that device size used in the cell may be specified by the designer depending on the requirement;
4. A symbol is provided for each cell that can be used in schematic editing. In a symbol, port type, such as digital or analog, of each connection is specified.

A specification file is a file to describe the cell library and the condition a hypermodel library to be created. Entries defined in this file include:

1. Hypermodel family. The hypermodel selected for insertion;
2. Config. The choice of hypermodel configuration;
3. Nominal voltage. The voltage for nominal operation of hypermodels;
4. Clock cycle. The period of time of clock signal for transient simulation;
5. Data file. The file that contains model arguments for cell templates. The data normally are provided by manufacturer of the cell library, i.e. model.sin;
6. Shm file. The name of hypermodel library file;
7. Power supply. The entry for specifying power supply node and value;
8. Signal level. The entry for specifying signal level and value;
9. Cell. The entry that define the cell that requires hypermodel to be inserted for simulation.

For each cell that may need a hypermodel to be inserted by the netlister, an entry must be included in the specification file. The format is shown below:

cell : <cell_name> : <device_size> : <port_list>;

where the first field is the entry identifier, **cell**.

The second field is the name of a cell. A template in the same name is available for simulator to access. The third field defines the device size. The fourth field defines the input and output pins of the template and the signals including power supply specifications.

A signal consists of sequence of signal segments. A segment may be a level or a transition. For a level segment, only one signal level, which is defined in either power supply entry or signal level entry, is needed and its value stays unchanged until next clock cycle.

For a transition, the format is level1->level2, where level1 and level2 are signal levels defined in signal level or power supply entries. For a transition, the signal starts at level1 at the beginning of a clock cycle. It changes to level2 in the middle and stays at level2 until next clock cycle.

An example of the library specification file is shown in Appendix A for information.

3. Hypermodel library generation

During design, a designer may choose a symbol of cell from the given cell library and place it on a schematic. ports of symbols are connected as needed. If a digital port is connected with an analog port, a hypermodel is to be inserted for simulation purpose [1].

A hypermodel is an interface model that couples a digital port and an analog port. There are three types of hypermodel including digital-to-analog, analog-to-digital and bidirectional and are named as d2a, a2d and bi, respectively.

Figure 1. illustrates how these hypermodels are inserted in a mixed-signal design.Note that a hypermodel simulates

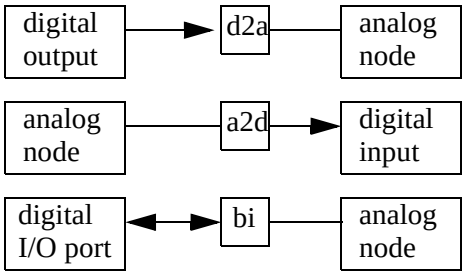

Figure 1. Connection of hypermodel

only the terminal characteristics of a digital port. All it does is the conversion of a digital signal into an analog signal or vise versa.

Since the terminal characteristics of a digital block vary with fabrication technology, i.e. cmos, bjt, etc., a family of hypermodels, d2a, a2d, and bi, are developed for a particular technology. For the cmos technology, a switch level hypermodel is proposed based on the current limited switch concept [2]. It appears the most cost effective and is used in the library generation.

The switch level d2a hypermodel can be illustrates by the block diagram shown in Figure 2.The input is a digital signal

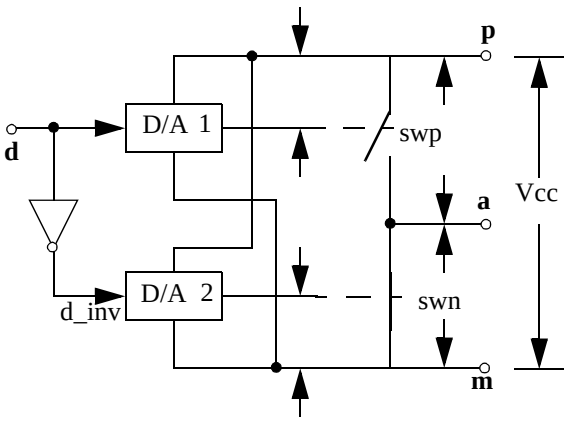

Figure 2. A switch level d2a hypermodel

that comes from a digital output port. This digital signal is applied at the control pin of a current limited switch. The upper switch, swp, is p-type. The lower switch, swn, is n-type. These switches correspond to the p-type and n-type mosfet in cmos technology, respectively. When the input digital signal is high, swp closes and swn open. The output is high.

On and off I_s~V_s curve of the current limited switch are shown in Figure 3. When a switch is off, its current is I_{off},

$$I_{off} = V_s / R_{off} \quad (1)$$

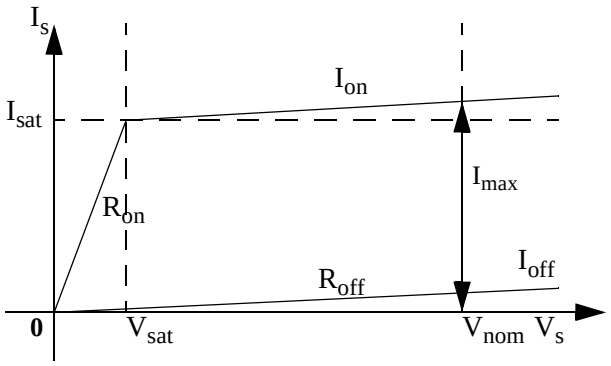

Figure 3. Switch I_s vs. V_s curve.

When the switch is on, its current is a piece wise linear function,

$$I_{on} = \begin{cases} V_s / R_{on} & V_s \le V_{sat} \\ I_{max} - (V_{nom} - V_s)/R_{off} & V_s > V_{sat} \end{cases} \quad (2)$$

where Vsat can be determined by the continuity condition,

$$V_{sat}/R_{on} = I_{max} - (V_{nom} - V_{sat})/R_{off} \quad (3)$$

The arguments R_{on}, R_{off}, I_{max}, etc. can be measured from mosfet channel current $I_{ds} \sim V_{ds}$ curve of a mosfet. I_{sat} can be calculated from the measured data.

With this switch level hypermodel, the hypermodel library generation process can be illustrated by the flow chart shown in Figure 4.

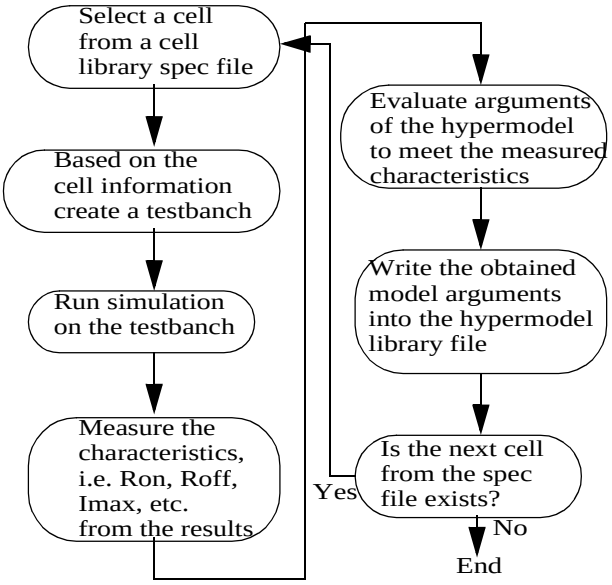

Figure 4. Hypermodel library generation process

For a selected cell library, the user provides a specification file that describe the required information about that cell library. See next section for the definitions of this specification file. Based on the information provided in the specification file, the testbanch and commands are created automatically to run DC, DT and TR analyses for measuring characteristics. The cell name and the obtained model arguments are written into the hypermodel library file according to the hypermodel library file format [3]. When the process is executed for each cell defined in the specification file, the hypermodel library file is closed and is ready for use by netlister. A default set of model arguments is also defined for any cell that is not listed in the hypermodel library file.

The entire process is implemented in AIM [4], a super set of Tcl/Tk scripting language [5].

4. Comparison of simulation results

To verify the characteristics of the characterized hypermodels, the same testbanch has been used for both the cell template and the hypermodel. The simulation results of a transient analysis on an inverter cell and that of the output of the switch level hypermodel, ids_d2an are shown in Figure 5.

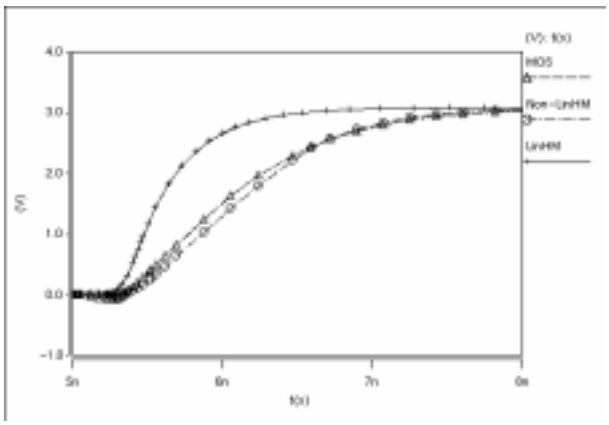

Figure 5. Comparison on simulation results

It can be seen that the transient response of the automatic generated hypermodel instance matches well with the response of the actual mosfet cell template.

5. Conclusion

The approach to automatically generate a hypermodel library file based on user's selection of cell library has been described. The only requirement to the user is to provide a specification file about the cell library. A generation program has been developed based on this approach.

The comparison on simulation results from a cell template and the characterized ids_d2an hypermodel shows that the characteristics matches reasonably well.

References

[1] M. Vlach: "Modeling and simulation with Saber", The 3rd Annual ASIC Seminar and Exhibit, Rochester, N. Y. Sept. 17-21, 1990.

[2] Ken G. Ruan, Jiri Vlach, J. A. Barby: “Logic Simulation with Current-Limited Switches”, IEEE Trans. on CAD, Vol. 9, No. 2, Feb 1990, pp. 133-141.

[3] “Mixed-Mode Interface”, Synopsys, Inc., 1989.

[4] “Customizing SaberDesigner Using AIM”, SaberDesigner Reference, Synopsys Inc. 2002.

[5] “Tcl/Tk Manual”, www.scripts.com/man. Tcl/Tk 8.0.5

Appendix A. A library specification file

This is an example of the library specifications file. Explanation are lines start with “#”. The format is

entry_id : entry_spec ;

where **entry_spec** varies with **entry_id**. The following is the entries and comments about each entry.

```
#1. Specify the family of hypermodel
hypermodel family : ids;
#2. Choose the available configuration of hypermodel
config : non_linear ;
#3. Specify operation condition
nominal voltage: 5.0;
clock cycle: 0.1u;
#4. Specify the file that contains the model arguments
data file: inc.sin;
#5. Specify the name of hypermodel library file
shm file: ids5.shm;
#6. Specify power supply and ground nodes
power supply : vdd : 5.0;
power supply : gnd : 0;
#7. Specify signal level to be used in signal definition
signal level : 0 : 0.0;
signal level : 1 : 5.0;
#8. List the cells that hypermodels are needed.
cell : inv1 : 2.0e-6 5u : in I  gnd->vdd, out O  vdd,
              p P  vdd, m G  gnd ;
cell : inv2 : 1.0e-6 3.0e-6 2.5e-6 3.0e-6 :
        in I  gnd->vdd, out O  vdd, p P  vdd, m G  gnd ;
cell : inv3 : 0.1e-6 5.0e-6 : in I  gnd->vdd,
              out O  vdd, p P  vdd, m G  gnd ;
cell : nand1 : 1.e-6 5.0e-6 : in1 I  gnd->vdd, in2 I  vdd,
              out O  vdd, p P  vdd, m G  gnd ;
cell : nand2 : 0.1e-6 5.0e-6 : in1 I  gnd->vdd,
              in2 I  vdd, in3 I  vdd, in4 I  vdd,
              out O  vdd, p P  vdd, m G  gnd ;
cell : dff : : ck I  0->1,  ckb I 1->0, d I   0 1,
              rb I  1->0,  q O  ,  qb  O  ;
```

Appendix B. An example shm file

This is an example hypermodel library file, which is generated by the hypermodel library generation program. This file is directly accessible by netlister for hypermodel insertion.

```
#
# This file is created for the ids hypermodel family.
#
# config: non_linear
# nominal voltage:  5.0
# data file: inc.sin
models {
idsdef:ids::
ids_tech..model      idsdef=();
inv1in:ids:idsdef:
  ids_tech..model  inv1in=idsdef<-(mode=non_linear,
      ronp=5802.9802760796,roffp=938.24992236884
      g,imaxp=351.58405080438u,ronn=2297.3692132
      726,roffn=999.99870295488g,imaxn=542.501569
      72557u,cop=21.337099545685f,con=21.3370995
      45685f,tt=565.41529503901p,cin=163.064877112
      73f,dvxh=2.5668558143607,dvxl=1.83340828398
      93,vnom= 5.0);
inv2in:ids:idsdef:
  ids_tech..model  inv2in=idsdef<-(mode=non_linear,
      ronp=11998.830713786,roffp=1.1258999068426t
      ,imaxp=166.75767255447u,ronn=3467.72018448
      67,roffn=997.33578601588g,imaxn=358.4227233
      6595u,cop=8.9675514329695f,con=8.967551432
      9695f,tt=400.99482679088p,cin=87.50559950575
      1f,dvxh=2.7685359515025,dvxl=1.695896737537
      3,vnom= 5.0);
dffin:ids:idsdef:
   ids_tech..model   dffin=idsdef<-(mode=non_linear,
      ronp=1578.9473684211,roffp=10t,imaxp=95u,ro
      nn=1943.3198380567,roffn=14.285714285714t,i
      maxn=123.5u,cop=1.9801325f,con=1.9801325f,tt
      =190p,cin=87.505599505751f,dvxh=2.768535951
      5025,dvxl=1.6958967375373,vnom= 5.0);
dffq:ids:dffin:
ids_tech..model dffq=dffin<-(ronp=1578.9473684211,
      roffp=10t,imaxp=95u,ronn=1943.3198380567,ro
      ffn=14.285714285714t,imaxn=123.5u,cop=1.980
      1325f,con=1.9801325f,tt=190p);
dffqn:ids:dffin:
     ids_tech..model      dffqn=dffin<-(ronp=2343.75,
      roffp=25t,imaxp=64u,ronn=3265.306122449,rof
      fn=33.333333333333t,imaxn=73.5u,cop=1.26298
      f,con=1.26298f,tt=190p);
}

pins {
:::          inv1in
inv1:::      inv1in
inv2:::      inv2in
dff:::       dffin ([4]=dffq, [5]=dffqn)
}
```