DEVELOPMENT OF CHIP MODEL LIBRARY FOR THE COMPUTER-AIDED ANALYSIS OF ELECTRONIC PACKAGES

S. N. Pratapneni, C. M. Wolff, P. Hsu, J. W. Rozenblit, and J. L. Prince

Center for Electronic Packaging Research
Department of Electrical and Computer Engineering
University of Arizona, Tucson, AZ 85721, U.S.A
Tel:(602) 621-8946, Fax: (602) 621-2999
e-mail: psatish@helios.ece.arizona.edu

ABSTRACT

This paper discusses the development of a chip model library for the complete design and simulation of Multichip Assemblies. The simulation of the driver/receiver part of the chips mounted in Multichip Modules(MCMs) can be done using a variety of models at varying levels of complexity. A hierarchy is used to organize the models for use with an intelligent model selection tool. Model selection is based on the tradeoff between accuracy and speed. The four models to be considered in this paper are device, table-based, equation based and simple RC models. The basic RC model, a physical model, considers transistor on-resistance and load capacitance as a complete representation of the driver circuit. The table lookup approach stores a detailed transfer function of device operation or circuit operation in a table using a device level circuit simulator. Equation-based models simplify the physical device equations based on the switching behavior of a particular circuit. An integral environment with this group of models is developed with an object oriented approach. Each of the model templates is treated as an object and it can be repeated as required.

1 Introduction

In order to achieve higher speed circuits, higher levels of integration at the board level are essential. As the complexity of packaging increases, complete design and simulation of such complex packages as multichip assemblies (MCA's) is absolutely essential. As multichip modules (MCM's) support higher levels of integration (merging with first and second level packages), the pad placement and chip-to-chip or chip-to-package connection in an MCM becomes crucial. In order to increase the efficiency and accuracy of the design and simulation, it is important to have a single environment under which a hierarchy of different models for ICs and packages are kept. Performance metrics (e.g., frequency, accuracy, computing speed etc.) can be used to select models for use in simulating a particular system product. For the simulation of large-scale chips and packages, highly efficient models are essential to achieve results in a time scale appropriate to the problem under consideration. So, providing a variety of model components and the capability to operate at several levels of complexity is very important.

The concept of a Chip Model Library (CML) is introduced in this work. The models used in the CML have been developed specifically for circuits used to drive or receive off-chip signals. These drivers and receivers can be simulated with a variety of methods at different levels of abstraction. On-chip driver circuits can be modeled either as models based on the actual physics of the component. For example, a device model of a transistor relates the current flow through the transistor to the terminal voltages, device topology, and manufacturing parameters of the actual device. This idea of a hierarchy of abstraction levels is useful when the tradeoff between simulation speed, accuracy and availability of manufacturing parameters is necessary. Very accurate delay predictions are possible with the standard circuit simulators like SPICE [1]. However, this approach is limited to circuits with no more than several thousand components because of the large computation time and memory requirements. The simplest modeling approach is to use RC models which are extracted from SPICE models. A second model, the lookup table approach, runs SPICE in advance of a transient simulation to calculate a set of parameters as a function of input and output voltages which are stored in tables [2]. Then, the tables are reused unless a process change has occured. When the required parameters are in between values in the table, interpolation is used at some loss of accuracy. The third approach is an equation-based model which improves upon previous work in this area [3, 4]

0-7803-1424-7/93 $3.00 ©1993 IEEE

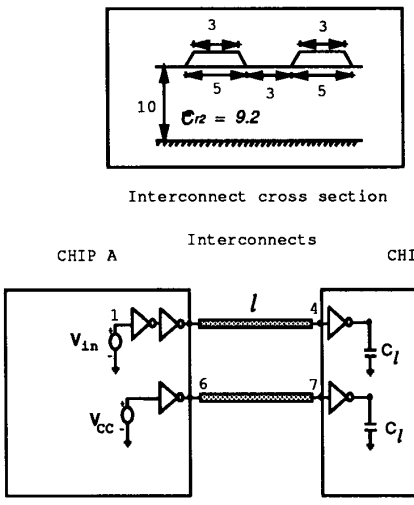

Figure 1: An example of MCM circuit simulated by using the CML and the PML.

by increasing the slew-rate range and reducing computation time.

The CML is a part of the Packaging Design Support Environment (PDSE) developed at the University of Arizona [5] to facilitate the analysis and design of packaging structures for microelectronic integrated circuits. Several independent tools have already been developed for interconnection parameter calculation and simulation. Present forms of interconnection simulators need manual entry of each and every transistor in the driver circuit [6], so an interface has been developed to facilitate automatic repetition of any driver component (e.g., an inverter or a NAND) and to communicate with existing interconnection simulators. The complete system simulation is accomplished by intelligent selection of the appropriate models from the CML. Each chip consists of an internal circuitry (i.e., core logic) and perimeter circuitry like drivers and receivers connected to the pad ring. As shown in Figure 1, the CML focuses on perimeter circuit modeling capability since existing tools are capable of modeling the internal chip functionality [7]. The user supplies specifications of a circuit such as a driver which the expert rules of the CML automatically translate to an appropriate model. These expert rules are based on driver model attributes.

An object-oriented approach [8] was used to develop this library (CML). Provided with the features of encapsulation, data abstraction, and inheritance, an object-oriented approach is suitable for the development of a hierarchical and modular data model with different levels of complexity [9]. The CML is flexible enough to incorporate new models with few changes to the program. The CML has been implemented using C++ and has been tested successfully. A driver circuit with a cascade of inverters was simulated using a modified SPICE model [6] and was interfaced with package models. The inclusion of the CML into the PDSE has helped it grow into a more complete analysis and simulation tool.

2 Model Description

2.1 Table Look-up Method

In order to get an approximate but faster simulation of digital circuits, many approaches have been invented. One such approach is table-lookup which is based on SPICE simulations. This approach can be implemented with a higher speed and greater flexibility if the device characteristics are measured extensively only once, then stored numerically into tables. The first table-lookup model was used in MOTIS, a fast timing simulator for digital MOS IC's [2]. A three dimensional(3-D) table of measured I_d for a set of (V_{gs},V_{ds},V_{sb}) was stored, and the stored points linearly interpolated to obtain I_d for intermediate voltages. The functional dependence as follows:

$$I_d = \mathrm{I}(V_{gs},V_{ds},V_{sb})$$

Assuming the source and body are connected together, then we can sweep the input V_{gs} and output V_{ds} voltages to get 2-D tables. These two tables are used for interpolation to get the output voltages. The load of each stage is computed using existing parameters. The β ratio of NMOS and PMOS are varied in a required range by keeping the first β constant while the other β is swept over the whole range and vice versa. Thus, if we divide the range of β by n, then $2(n+1)^2$ tables are formed. Once formed, new tables are not needed unless process changes occur. It was anticipated that since table-lookup should be more efficient than SPICE, the computation time for a chain of inverters would be reduced drastically as the number of stages increases compared to SPICE. However, it was found that as number of stages increases the gain of CPU time over SPICE first rises, peaks, and then falls as shown in Figure 2.

2.2 Equation-based method

Standard circuit simulators like SPICE can be used for transient analysis and delay evaluation in logic systems; but, with increasing number of transistors

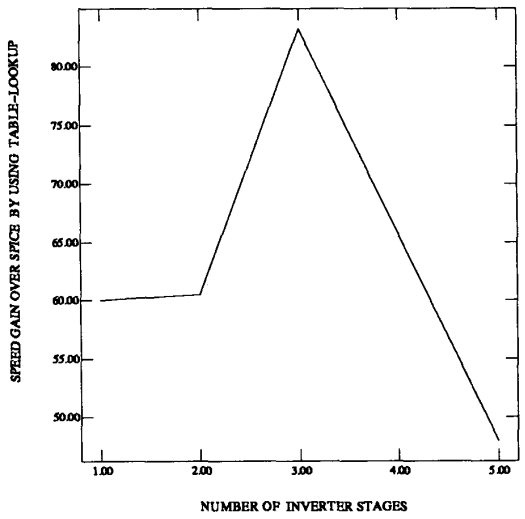

Figure 2: Figure showing gain in computation time over SPICE by using table-lookup approach versus number of stages.

in VLSI circuits, the evaluation of delay using SPICE requires an enormous amount of memory space and CPU time. In order to make simulation faster, many approaches have been developed in past, such as using tables for the device equations [2], relaxation methods for solving the differential equations [10], and macromodelling [11]. In addition, analytical expressions have been developed by many authors to evaluate the propagation delay, rise and fall times of the basic CMOS logic gates [3, 4, 12]. These models are a tradeoff between accuracy and speed of computation because simplifying assumptions are used such as neglecting the shape of the input waveform, using approximations for load currents [12], or neglecting the PMOS(NMOS) currents in the falling (rising) output delay evaluations.

A modified equation-based(E-B) model of a CMOS inverter has been developed for the CML that works with greater efficiency than previous techniques, while taking into account the shape of the input waveform, the capacitive load, and the transconductance of the PMOS and NMOS transistors. This E-B method utilizes a combination of simple characteristic equations of transistors and a power series approximate solution for the differential equations that govern the behavior of the inverter. The regression equation has been developed for the time spent in different regions of operation to help improve the speed of the program. Comparison of delay times is also made along with the accuracy of the output of the various models. An additional advantage is an improvement in computation time because of the moderate amount of computations per region.

Unlike the previous work [3], both PMOS and NMOS currents are considered in the saturation region in this model to increase the accuracy of the model. For a rising input in the linear region, the PMOS(NMOS) current is neglected without significant loss of accuracy while reducing the computation time. This approximation is also made in the simplified RC model.

2.3 Simplified RC model

A simplified RC model for CMOS drivers as is commonly used in existing logic simulators such as RSIM [7] is also used in the CML. It greatly reduces the computation time at the expense of accuracy. The transistors have been replaced by the on-resistance and the gate capacitance calculated using technology parameters [13]. Further, at any stage only one of these resistances is used: pull down(NMOS) during discharging and pull up(PMOS) during charging.

In these simplified models interstage capacitances are calculated using existing parameters. Both input and output capacitance(C_{in} and C_{out}) of each stage are computed [13] as follows

$$C_{in} = C_{oxn}\ W_n + C_{oxp}\ \beta W_n L_p + C_{oxn} W_n LD_{sn} + C_{oxp}\ \beta W_n LD_{sp} + C_{oxn} W_n LD_{dn} + C_{oxp}\ \beta W_n LD_{dp}$$

$$C_{out} = (C_{eqdbn}\ AD_n + C_{eqdsn}\ PD_n) + (C_{eqdbn}\ AD_p + C_{eqdsn}\ PD_p)$$

where $AD_n = W_{dn}\ L_{dn}$, $AD_p = W_{dp}\ L_{dp}$, $PD_n = 2(W_{dn} + L_{dn})$, and $PD_p = 2(W_{dp} + L_{dp})$. AD is defined as the area of the drain diffusion, and PD is defined as the perimeter of the drain diffusion region.

3 Model Development

As illustrated in Figure 3, the Chip Model Library(CML) is organized as a hierarchical structure.

The object-oriented approach was applied to manage the complexity of the CML. In the world of object-orientation, individual data groups are naturally encapsulated as separate objects. Objects are instances of certain classes. The interdependencies among objects include decomposition and specialization. Decomposition means an object is split into components. Each component is itself an object. For

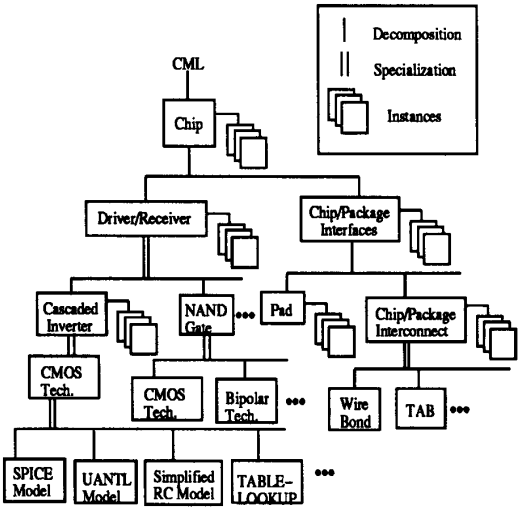

Figure 3: A structural representation of the CML.

example, a chip object in the CML can be decomposed into drivers/receivers and chip/package interfaces which are in turn different objects.

Specialization is a taxonomic representation for the kinds of variants that are possible for an object (i.e., how a more general object can be categorized and subclassified). A more general object contains a set of attributes or variables that are common to all alternate implementations of a given function. For instance, common attributes of a driver would include power dissipation, delay, area, and on-resistance. A driver can further be classified as a *specific* inverter, NAND gate, or AOI gate depending on the different structures and applications required. A NAND gate, for example, can be further categorized into CMOS or Bipolar driver based on the technologies available for implementation. Each CMOS driver (e.g., cascaded inverter) is in turn defined by lower level categories along with an associated model. The general-to-specific relations within the hierarchical structure of the Chip Model is realized through the inheritance mechanism provided by the object-oriented programming language C++. Because of this mechanism, new classes (specialized classes) can be derived from base classes. The derived class inherits all the data structure slots and functions of the base class and can define new slots and member functions of its own.

In Figure 4, a section of C++ code illustrates how the object-oriented approach can be applied to construct a UANTL CMOS cascaded inverter from PMOS and NMOS objects.

```
class PMOS
{
  protected:
  int Nc;
  int Npm;//PMOS transistor number
  int Nd, Ns, Nb, Ng;//node number
  PMOS();//constructor
  //UANTL mos parameter
  public:
  float vto, gamma, k, lambda, phi, cgd, cgs, cgb,is
            mj, cbd, cbs, pb, dy;//dummy
  void input(Window,int,int);// input device parameters
  void setnode(int,int,int,int);
  void setNo(int n){ Npm=n;}
  .....
}

class NMOS: public PMOS //derive from class PMOS
{
  protected:
       int Nc;
       int Nnm; //NMOS transistor number
       NMOS();// constructor
}

class CMOS_INV // a CMOS inverter is composed of two objects
{
  public:
  PMOS mypmos;
  NMOS mynmos;
};

class CASCADED_CMOS_INV
{
  private:
  int Nstage;
  float ratio;
  public:
  CMOS_INV *mycv;
  CASCADED_CMOS_INV();
  void input_stage(Window);
  void input_ratio(Window);
  void generate_INV();
  void construct(Window);
  int get_stage(){return Nstage;}
  float get_ratio(){ reurn ratio;}
};
```

Figure 4: The implementation of the CML using object-oriented approach.

In this example, NMOS is a specialized class of its base class PMOS. Decomposition is illustrated by the CMOS inverter class which is composed of PMOS and NMOS objects as its members. Note that using object-oriented approach, we reduce the complexity of modeling, increase the software reusability, and, thus, simplify the programming.

The entry of a new chip model description begins with an empty hierarchy that is filled with attributes and model parameters. The underlying representation scheme for a chip model stored in the CML is similar to a frame-based knowledge representation. Note that a chip model frame contains slots for thermal and physical properties of a chip along with the electrical models of the nodes lower down in the hierarchy so that both electrical and thermal/mechanical

ChipName:mychip1
Thermal Slot:
R_{ja}: junction to ambient thermal resistance
R_{jh}: junction to header thermal resistance
P_d: power dissipation, W
S_p: sensitivity of junction to ambient
Physical Slot:
dx: x dimension of the chip
dy: y dimension of the chip
Electrical Slot:
CMOS
Cascaded Inverter(UANTL)
stage: 1
ratio: 1
PMOS parameters
vto gamma k lambda phi cgd cgs cgb
-1,0.37,0.005,0,0.3,2.41e-15,2.415e-15,0
is mj cbd cbs dy pb
1e-15,0.33,1.47e-14,1.58e-14,0,0.6
NMOS parameters
vto gamma k lambda phi cgd cgs cgb
1,0.37,0.005,0,0.3,2.41e-15,2.415e-15,0
is mj cbd cbs dy pb
1e-15,0.33,1.47e-14,1.58e-14,0,0.6

Figure 5: A frame based representation of a chip model.

simulations can operate with the same hierarchy. A typical chip model frame is shown in Figure 5.

4 Case Study

In this section, the simulation of a transmission line system with the circuit simulator UANTL demonstrates how the CML can be applied to facilitate the analysis of electronic packaging. As shown in Figure 1, it is assumed that chip A (driving chip) and chip B (receiving chip) are two of the chips on a multichip system such as an MCM. The off-chip interconnect models (transmission lines and substrates in this example) are provided from the other library called the Package Model Library (PML) [14] which is beyond the scope of this paper. The PML includes the package from the bondwire to the board (e.g., vias, transmission lines, and boards). The termination networks of the transmission line system for simulation are retrieved from the CML in a predefined syntax such as *ChipA.pad1.inverter(UANTL) connection node 1, connection node 2* in our simulation environment. Note that our goal is to construct a "flattened" circuit from the hierarchical structure of the CML so that the netlist can interface with a common circuit simulator such as UANTL or SPICE. The UANTL

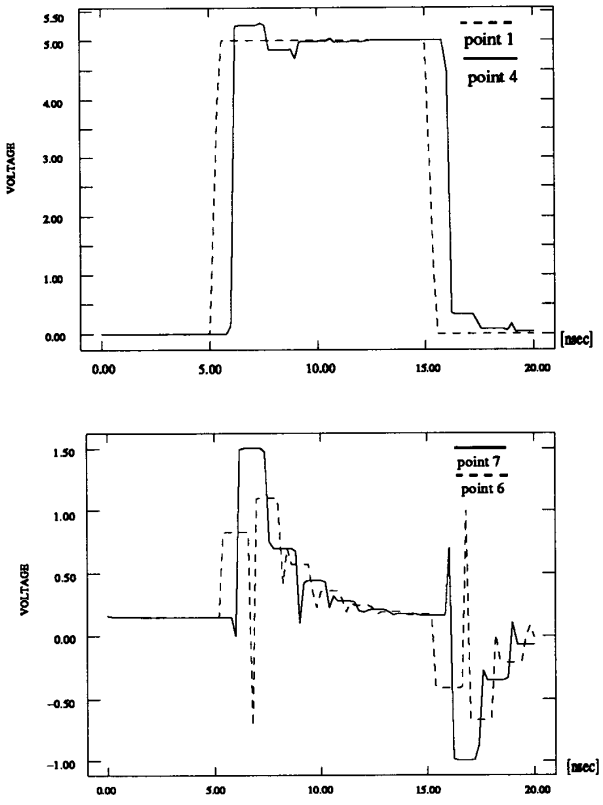

Figure 6: Simulation results of the transmission line system in Figure 1.

simulation results of the circuit in Figure 1 are shown in Figure 6. In this case, the rise time/fall time of the input signal is 0.5ns and the output is observed at the far end of the active line. The crosstalk noise can be observed at both the near end and far end of the quiet line.

5 Conclusion and Future work

This work was a successful step towards integration of package and chip simulation. A common platform for modeling and simulation of system has been presented and it was supported by an example of an MCM. In future work, further integration of different models for drivers/receiver will be pursued.

6 Acknowledgements

This research is supported in part by Semiconductor Research Corporation. We also want to thank Mr. A.

Vaidyanath and Mr. B.Thoroddsen for their useful support and conducive discussions regarding table-lookup approach.

References

[1] B. Johnson, T. Quarles, A. R. Newton, D. O. Pederson and A. Sangiovanni-Vincentelli, SPICE3 Version 3D2 User's Manual, Department of Electrical and Computer Sciences, Univ. of California, Berkeley, Oct. 31, 1990.

[2] B. R. Chawla, H. K. Gummel and P. Kozak,"MOTIS-An MOS timing simulator", *IEEE Transactions of Circuit Theory*, vol. 20, no. 6, pp. 901-910, June, 1975

[3] S. R. Vemuru and A. R. Thorbjornsen, "A model for delay evaluation of a CMOS inverter ", in *1990 IEEE International Symposium on circuits and systems*, Vol. 1, pp- 89-92, Urbana, New Orleans, LA, May, 1-3, 1990.

[4] Young-Hyun Jun and Ibrahim. N. Hajj, "An efficient timing simulation approach for CMOS digital circuits", *Proceedings of the 33rd Midwest Symposium on circuits and systems*, Vol. 1, pp. 235-8, Calgary, Canada, Aug. 12-14 1990.

[5] J. W. Rozenblit, et al., "Computer Aided Design System for VLSI Interconnections",*Proc. of 1989 IEEE Intl. Conf. on Computer Design: VLSI in Computer and Processors*, pp. 237-241, 1989.

[6] J. C. Liao, O. A. Palusinski, and J. L. Prince, "Computation of Transients in Lossy VLSI Packaging Interconnects," *IEEE Trans. on Components, Hybrids, and Manufacturing Technology*, vol. 13, No.4., p.833, Dec. 1990.

[7] Christopher J. Terman, "Timing Simulation for for Large Digital MOS Circuits", *Advances in Computer-Aided Engineering Design*, Vol. 1, pp. 1-92.

[8] Bjarne Stroustrup, *The C++ Programming Language*, Chapter 12, Addison–Wesley Publishing Company, New York, 1991.

[9] G.Booch,"Object-Oriented Design with Applications",Benjamin/Cummings, Mento Park, CA, 1991.

[10] E. Lalarasmee, A. E. Ruelhi and A. L. Sangiovanni Vincentelli, "The waveform relaxation method for the time doma in analysis of large scale integrated circuits", *IEEE Transactions on Computer Aided Design*, vol. CAD-1, no. 3, pp. 131-145, July 1982.

[11] M. D. Matson and L. A. Glasser, "Macromodelling and optimization of digital MOS vlsi circuits", *IEEE Transactions on Computer Aided Design*, vol. CAD-5, no. 4, pp. 659-678, Dec. 1986.

[12] D. Deschacht, M. Robert and D. Auvergne,"Explicit Formulation of Delays in CMOS Data Paths", *IEEE Journal of Solid-State Circuits*, Vol SC-23, no. 5, pp. 1257-1264, October 1988.

[13] A. J. Al-Khalili, Y. Zhu and D. Al-Khalili, "A Module Generator for Optimized CMOS Buffers", *IEEE Transactions on Computer-Aided Design*, Vol. 9. No. 10, pp. 1028-1046, October 1990.

[14] P. Hsu, J. W. Rozenblit, S. N. Pratapneni and C. M. Wolff, "An Integrated System for Design Automation of VLSI Interconnects and Packaging", to be presented in *36th IEEE Midwest Symposium on Circuits and Systems Conference*, Detroit, August 16-18, 1993.