An Integrated System for Design Automation of VLSI Interconnects and Packaging

P. Hsu, J. W. Rozenblit, S. N. Pratapneni, and C. M. Wolff, 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) 621-8076

Abstract — The packaging design support environment (PDSE) is a software system being developed at the University of Arizona to facilitate the analysis and design of packaging structures for microelectronic integrated circuits, a subject which is becoming one of increasing importance with higher circuit integration and system performance. PDSE provides a platform for work in several active research areas including interconnect and packaging modeling and simulation in electrical, thermal, and thermal-mechanical aspects, CAD framework development and evaluations for performance, manufacturability, and reliability, etc. This paper describes the overall architecture and characteristics of the PDSE system in development, its implementation and applications.

INTRODUCTION

Advances in high-speed, high-density digital interconnects have brought packaging to the top of the list of electronic system performance-limiting factors. With today's small feature sizes and high levels of integration, the speed of on-chip circuit is so fast that a significant portion of the total delay in a high-performance electronic system comes from the chip-to-chip delay and packages. In order to minimize the delay, interconnect length is reduced, chips are packed closer. However, these efforts brought new problems such as crosstalk, reflection, and delay due to the associated transmission line effects. The large amounts of heat generated by such high-speed and densely packed circuits have to be removed efficiently also.

In order to design manufacturable packaging structures for high-speed, high-density circuits and leading-edge electronic systems, it is essential to be able to evaluate and predict the electrical phenomena prior to the actual design implementation. With complexities and packing densities which exist today, the practical way to perform this evaluation with confidence is by use of modeling and simulation techniques. It is crucial that sophisticated computer-aided tools be available to support design automation of VLSI packages.

Starting from 1984, several CAD tools for analysis of microelectronic interconnects/packages have been developed at the Center for Electronic Packaging Research at the University of Arizona for aforementioned purposes. The overview of the theoretical backgrounds and simulation techniques for the developed software tools can be found in [1, 2]. There are four major groups of tools currently in use: electrical parasitic extraction tools, transmission line circuit simulators, thermal analysis tools and thermal-mechanical analysis tools. This paper discusses how these tools are structured and integrated into a useful simulation environment for design automation for VLSI interconnections and packaging. The overall architecture and characteristics of the PDSE system will be described with emphases on CAD tool integrations and design data management.

In the next section, previous effort of the PDSE is reviewed. We then describe a framework based on design data representation and management. To support first level (chip level) and second level (board level) packaging, two major libraries called the Chip Model Library (CML) and the Package Model Library (PML) are established. We describe how simulation models are generated from these data base and then interfaced with the aforementioned CAD tools for design verifications. An illustrative example is given to demonstrate the capability of the integrated system for design automation of VLSI interconnects and packaging. In the last section, we conclude this paper and address the future work.

BACKGROUND

The early work of the PDSE [3] emphasized on interconnect/packaging CAD tool integrations and simulation management. The X-Window based common graphical user interface (GUI) provided in the PDSE includes pop-up menus, dialog boxes, spread-sheet windows, and plotting windows. The previous framework was implemented in C programming language and running on top of the UNIX based workstations. Users can activate different tools through selection menus. Design data models for simulation are tightly coupled to specific tools used. There is no unified representation of design data in this scheme. This increases the overhead of maintaining different design database as the number of CAD tools integrated increases for more complex packaging simulations. Simulation management in the PDSE includes design constraint checking, error handling, post processings and design ranking through a scheme called experimental frame [3, 4]. PDSE allows up to ten variations of package models (transmission-line systems), and ten different experimantal frames to be simulated at the same time. The experimental frame is defined independently of the models. It consists of a set of input waveforms for each voltage source defined in the model and control variables (simulation time period and a set of maximum allowable voltages for each probe point in the circuit). Once a set of models and experimental frames have been defined, the simulation process may begin. Using this scheme, electrical performance of different models can be compared for different inputs. The transient responses of the probe points in the circuits for different combinations of experimental frames and models can then be displayed on the PDSE post processing window system for design trade-off study and ranking of alternatives.

THE FRAMEWORK

As shown in Figure 1, the underlying architecture of the current PDSE consists of three major layers: design data base, CAD tool encapsulations and simulation management. The design data base and CAD tool encapsulations are realized in an object-oriented approach using C++ programming language. It is es-

CH 3381-1/93/$01.00 ©1993 IEEE

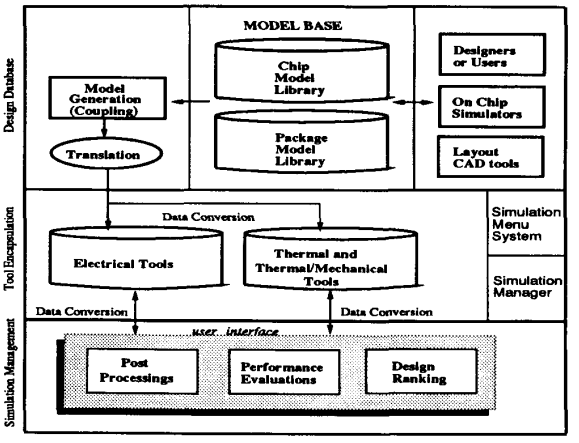

Figure 1: The architecture of PDSE

sential to mix the current C++ implementation with the existing C language based support environment. We reuse the common GUI introduced in the previous section which is implemented in X11R4. The design data base layer consists of two major libraries termed the Chip Model Library (CML) and the Package Model Library (PML). The CML includes physical, thermal (under development) and electrical (especially peripheral termination network models such as drivers, receivers, etc.) information of chips with different levels of complexity and abstraction. The PML includes models for multilevel (up to board level) interconnect structures (discontinuities such as bends, vias, and crossovers, etc. are also included). Automatic model generation for simulation requires effective interface and coupling between these two major libraries. Note that the development of the CML and the PML is to provide a common (unified) database to support both electrical and thermal/mechanical simulations. Each CAD tool is encapsulated as an object in the CAD tool encapsulation layer of the PDSE. Most of the UA tools were written in FORTRAN programming language and applied different numerical methodologies. In each tool object, several methods are usually provided as translators. The performance of file conversions is twofold (as shown in Figure 1). One conversion is between the real software tools and the simulation models generated from the PML and the CML. The other major conversion is between the output data generated from execution of the tools and postprocessing programs (or another CAD tool). All these details are encapsulated in an software object and transparent to the users. The third layer of PDSE is for post processings of the simulation results such as graphical displays or error handlings. Performance evaluation of the simulated package structure is also proceeded in this layer.

The Chip Model Library

Development of the CML is a crucial step towards achieving the final goal of developing a complete package analysis and simulation tool. The simulation of the driver/receiver part of the chips mounted in a multichip system (e.g., a board level package) 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. 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. The accuracy desired in different simulations is different, so their is no need for using models of same accuracy for all the cases. So, providing a variety of model components and the capability to operate at several levels of complexity is very important. Model selection is based on the tradeoff between accuracy and speed. The four models included were 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. It greatly reduces the computation time at the expense of accuracy. Further, at any stage only one of these resistances is used: pull down (NMOS) during discharging and pull up (PMOS) during charging.

The table lookup approach stores a detailed transfer function of device operation or circuit operation in a table using a device level circuit simulator. The drain current of each transistor is dependent on gate to source voltage, V_{gs}, and drain to source voltage, V_{ds}. By sweeping these two voltages 2-D tables were formed. These two tables are used for interpolation to get the output voltages. The load of each stage is computed using existing parameters.

Equation-based models simplify the physical device equations based on the switching behavior of a particular circuit. Behavioral models are usually only designed to produce the correct timing delay of a block, it will likely not predict switching noise properly. A modified equation-based(E-B) model of a CMOS inverter has been developed and used for the CML that works with greater efficiency than previous techniques [6, 7], while taking into account the shape of the input waveform, the capacitive load, and the transconductance of the PMOS and NMOS transistors.

Each chip consists of an internal circuitry like drivers and receivers connected to the pad ring. The CML focuses on perimeter circuit modeling capability. Since other tools [9] are capable of modeling the internal chip functionality.

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.

As illustrated in Figure 2, the CML is organized as a hierarchical structure.

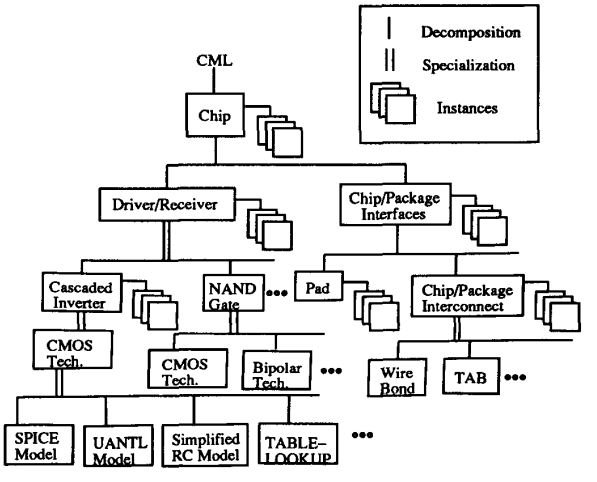

Figure 2: A structural representation of the CML.

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 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.

The Package Model Library

The Packaging Model Library encompasses models for different types of boards, modules (or just called substrate when a multichip module is concerned) and multilevel interconnects. The structural representation of the PML is shown in Figure 3. It can be further decomposed into four sublibraries: discontinuities, conductors, substrates and boards. The implementation of the PML is similar to that of the CML. However, we concentrate on off-chip interconnect structures among and ouside of the chips.

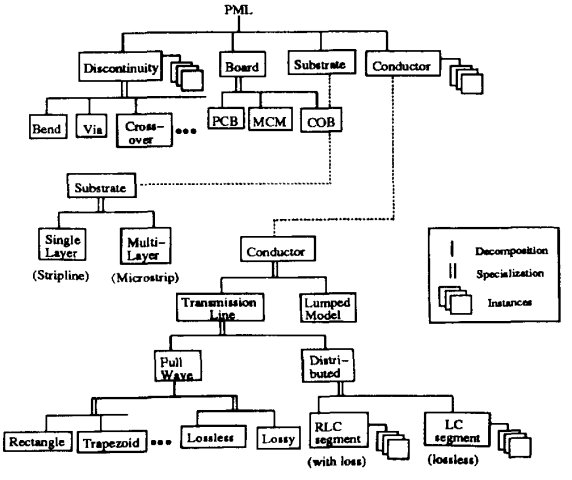

Figure 3: A structural representation of the PML.

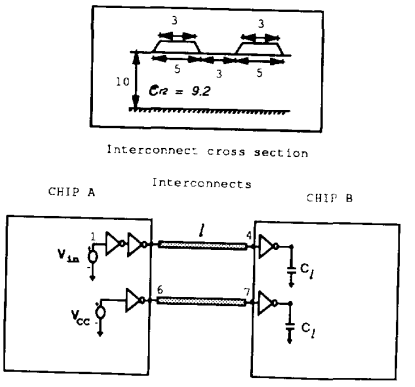

Figure 4: An example of multichip circuit simulated by using the CML and the PML.

Simulation Model Generation

Simulation models are generated from coupling between the PML and the CML. Two coupling schemes are provided in the current PDSE: coupling between chips and packaging (e.g., chips on a printed circuit board) and chip to chip coupling (e.g., an off-chip interconnects with termination circuitry on the chips.) The coupling and retrieval of models are achieved from specifying a high level design file called Packaging Description File (PDF). The PDF specifies chip placement on a board and chip to chip interconnects. Use of PDF to generate models for simulation is illustrated in the next section.

APPLICATIONS

In this section, one of the generated models through the PML and the CML for transmission line simulation is used as an illustration. As shown in Figure 4, it is assumed that chip A (driving chip) and chip B (receiving chip) are two of the chips on a multichip system such as a multichip module. The off-chip interconnect models (transmission lines and substrates in this example) are provided from the PML. A fragment of PDF to generate models for simulation is shown in Figure 5. Note that in this example, the goal is to construct a "flattened" circuit from the hierarchical structure of the CML and the PML so that the netlist can interface with a common circuit simulator such as UANTL [5] or SPICE. UANTL is a lossless transmission line simulator and the abstraction level of the driver/receiver is specified as "UANTL" in the PDF in Figure 5. The UANTL simulation results of the circuit in Figure 4 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.

CONCLUSION

An integrated framework for design automation of VLSI interconnects and packaging has been presented. Based on the software tools developed at the University of Arizona, we introduced a new scheme for CAD tool integration, modeling and simulation for electronic packaging using an object-oriented approach. Simulation model generation through a centralized data base

(libraries) has been demonstrated. A simulation example for transmission line analysis was given.

ACKNOWLEDGEMENT

This research is supported in part by Semiconductor Research Coorporation under contract 91-MP-086.

References

[1] J. L. Prince, O. A. Palusinski and A. C. Cangellaris, "Modeling packaging characteristics for high speed devices", *Journal of semicustom ICs*, Vol.8, No.4, pp.30–38, 1991.

[2] James E. Morris, Electronic Packaging Forum, Vol. 1, Chapter 4, Van Nostrans Reinhold, New York, 1990.

[3] J. W. Rozenblit, et al., "Computer Aided Design System for VLSI Interconnections", *Proc. of 1989 IEEE International Conference on Computer Design: VLSI in Computer and Processors*, pp.237–241, 1989.

[4] J. W. Rozenblit, "A Conceptual Basis for Integrated, Model–Based System Design", Technical Report, University, 1986.

[5] O.A. Palusinski, J. C. Liao, J. L. Prince, A. C. Cangellaris, "Simulation of Transients in VLSI Packaging Interconnections", *IEEE Trans. on Components, Hybrids, and Manuf. Tech.*, Vol. 13, pp. 160–166, 1990.

[6] 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.

[7] 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.

[8] S. N. Pratapneni, C.M. Wolff, P.Hsu, J.W Rozenblit, J.L Prince, "Development of Chip Model Library for the Computer Aided Analysis of Electronic Packages", accepted for publication in *IEEE/CHMT International Electronics Manufacturing Technology, Symposium*,(accepted for publication), October 4-6, 1993 Santa Clara, CA.

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


```
CHIPS:
  begin
    chipA chipB
  end
SUBSTRATES:
  begin
    2Layer
  end
TRANSMISSION_LINE_SYSTEM:
  begin
    DRIVERS:
      begin
      chipA.pad1.inv(UANTL) 1 3
      chipA.pad2.inv(UANTL) 6 8
      end
    RECEIVERS:
      begin
      chipB.pad1.inv(UANTL) 4 5
      chipB.pad2.inv(UANTL) 7 9
      end
INTERCONNECTS:
    begin
      my2Line
    end
  end
```

Figure 5: A fragment of PDF for a transmission line simulation.

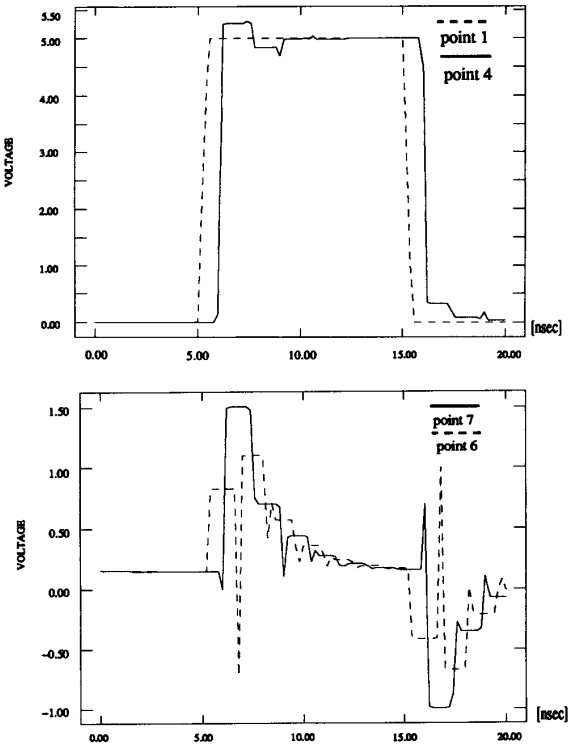

Figure 6: Simulation results of the transmission line system in Figure 4.