ELSEVIER

INTEGRATION, the VLSI Journal 17 (1994) 163–187

INTEGRATION
the VLSI journal

A computer-aided design framework for modeling and simulation of VLSI interconnections and packaging ★

Pochang Hsu *,
Advanced Package Development, LSI Logic Corporation, MS K-300, Fremont, CA 94539, USA

Jerzy W. Rozenblit
Center for Electronic Packaging Research, Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721, USA

Received 15 April 1994; revised 14 July 1994

Abstract

The higher speed requirement and rising complexity of interconnect and packaging structure in a VLSI system have increased the necessity of applying modeling and simulation techniques to develop CAD tools for analysis and design. To effectively manage design data and CAD tools involved for modeling and simulation of electronic packaging, a framework which provides different levels of services and abstractions is essential. This paper describes a computer-aided design framework which provides three levels of services for the aforementioned purposes. The first level of the framework supports CAD tool integrations and simulation management. A common graphical user interface is provided for the simulation environment. In the second level, design data representation and management are stressed. We applied an object-oriented approach to develop design libraries and encapsulate CAD tools. The third level of the framework emphasizes system level modeling and simulation for multiple chip systems. The underlying architecture and implementation of the framework are explained, design examples given.

Key words: CAD framework; Electronic packaging; Simulation; Tool integration

1. Introduction

Electronic packaging is one of the most important performance-limiting factors for today's high-density and high-speed integrated circuits. With today's small feature sizes and high levels of inte-

★ This work was supported in part by the Semiconductor Research Corporation through the electronic packaging program at the University of Arizona, Tucson. Part of this paper was presented at the 1st IEEE Topical Meeting on Electrical Performance of Electronic Packaging, April 22–24, 1992 and 36th IEEE Midwest Symposium on Circuits and Systems Conference, August 16-18, 1993.

* Corresponding author.

SSDI 0167-9260(94)00010-7

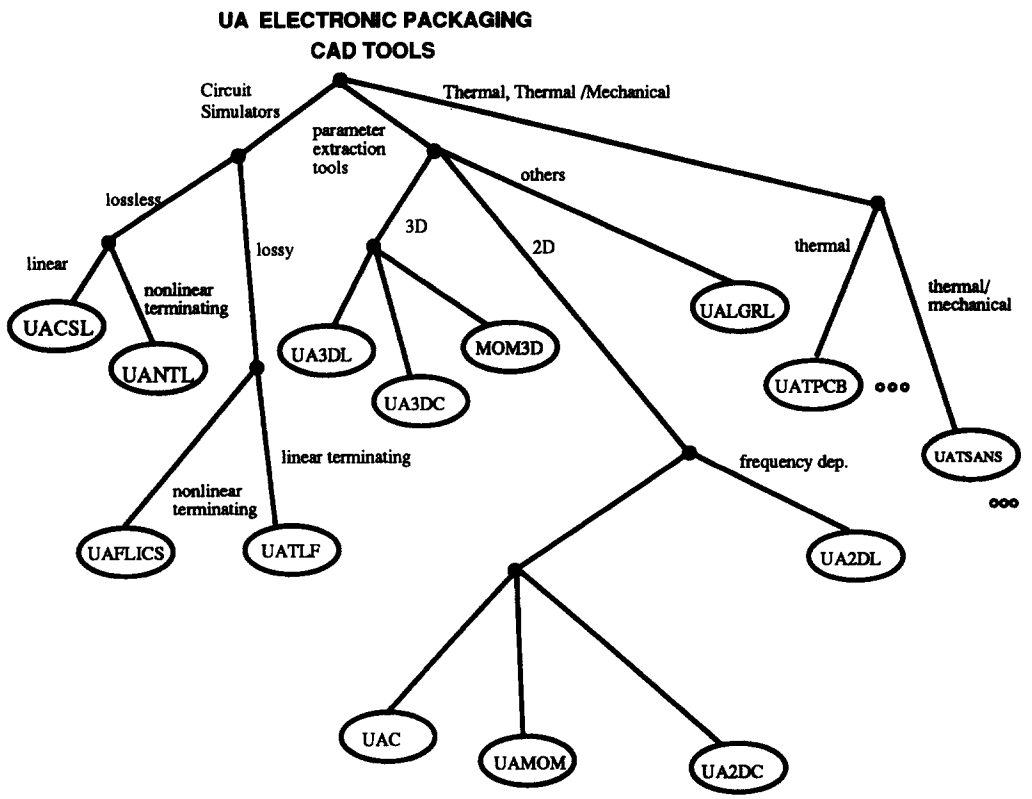


Fig. 1. Electronic packaging software tools developed at the University of Arizona.

gration, the speed of the on-chip circuitry is so fast that a significant portion of the total delay in a processing unit comes from the chip-to-chip delay and packages [1]. In order to minimize the delay, interconnect length is reduced, devices are scaled down and chips are packed closer. However, these efforts bring new problems such as crosstalk, reflection, and delay due to the associated transmission line effects. High-speed and densely packed circuits generate large amounts of heat that have to be removed efficiently. The capability to analyze and verify a complex package/interconnect design before physical implementation of assemblies is critical to assure the proper functionality and manufacturability of a system design. Simulation and modeling techniques have been applied and CAD tools developed to perform complex packaging/interconnect analysis with high confidence. Many CAD tools have been developed at the Center for Electronic Packaging Research at the University of Arizona for analysis of electronic packaging and interconnects since 1984. These tools include electrical parasitic extractors, transmission line circuit simulators, simultaneous switching noise simulator, board level thermal analysis tools and thermal-mechanical stress/strain analysis tools. Part of the developed CAD tools are classified in a hierarchical manner according to their functionalities and shown in Fig. 1. [2,3] are good sources for more detailed descriptions of the developed tools.

Since each tool was designed individually by different developer in different periods of time, to facilitate the use of these CAD tools for problem solving, effective management of design data and integration of heterogeneous tools for modeling and simulation become another critical issue. This usually relies on a successful computer framework which provides different levels of services. A

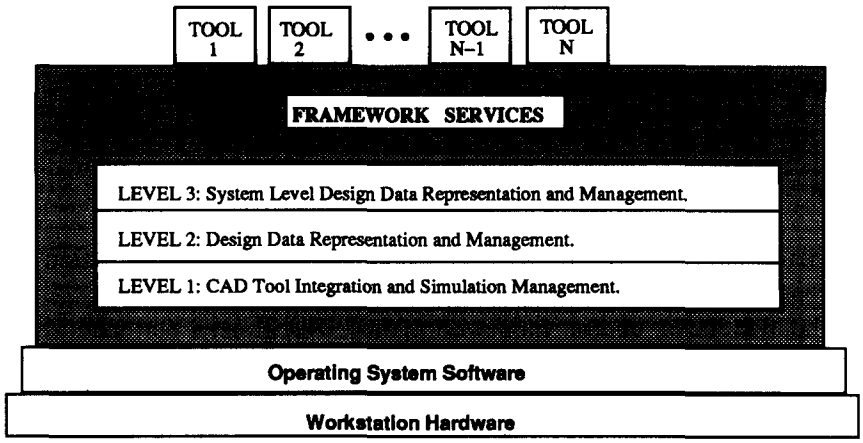


Fig. 2. The architecture of a three layered CAD framework for electronic packaging.

CAD framework [4] represents a collection of mechanisms or facilities (programming libraries, extension languages, data management and user interface facilities, and so on), at many different levels of abstraction, that are, to varying degrees, specific to the electronic CAD world. An ideal CAD framework which fulfils different requirements (accuracy, computational speed, lower or higher level designs, etc.) for a package or a system designer should provide design models and simulation CAD tools with different levels of abstraction: from lower levels of abstraction which are closer to the manufacturing process to higher levels which are closer to a designer's conceptualization.

In this paper, we describe a framework which is used to configure a set of packaging analysis tools and to develop interfaces to support model generation, simulation and design verification for the first level (a single chip) and the second level (multi-chip packages or modules) packaging structures. The framework is to support both electrical and thermal/mechanical analyses for electronic package on the same platform. This framework, as shown in Fig. 2, based on the CAD tools developed at the University of Arizona, provides three levels of services: (1) CAD tool integration and simulation management, (2) design data representation and management, and (3) system level modeling and simulations.

The presentation of this paper follows the three layered architecture of the framework. In the next section, integration of the tools and the management of simulations are discussed. A simulation environment called the Packaging Design Support Environment (PDSE) is implemented. We then focus on the issue of design database management for electronic packaging. To support model generation for both electrical and thermal/mechanical simulations, two major libraries termed the Packaging Model Library (PML) and the Chip Model Library (CML) are developed using an object-oriented approach to implement different levels of model abstraction. The building blocks in the two libraries can be used to partition a board (or module) level packaging structure. Model generation for simulation and analysis is made possible by effective coupling/interfacing between these two major libraries. In this layer, we introduce a different scheme for CAD tools integration using an object-oriented approach. Each tool is encapsulated as an object clustering around the CML and the PML. Section 4 presents the layer of framework for system level packages such as MCMs (multichip modules). This layer provides a global view of analysis instead of a detailed simulation. Examples utilizing resources in different layers of the framework described in this paper are given in Section 5. A CMOS based four chip MCM is used for both electrical and thermal analyses. Related

work of the described framework is surveyed in Section 6. The usefulness of developing such a framework is addressed. Section 7 concludes our research.

2. Tool integration and simulation management

This section presents our early effort for CAD tool integration and simulation management. A simulation environment called the Packaging Design Support Environment (PDSE) implemented the core concepts of the first layer of our framework [5,6]. PDSE is a software shell and the tool integration in this layer of framework is based on a common graphical user interface implemented in X11R4 to provide the interactions between users and packaging simulators. The integration mechanism is very tool dependent. Whenever there is a new tool to be integrated in the software shell, necessary input/output menus for manipulating the tool are created and then translators between the software shell and the software tools are provided. In this manner, PDSE is able to drive every packaging simulator.

The design process in PDSE proceeds in three major phases: modeling, simulation and evaluation. These processes are interactive and allow a package designer to refine a design model, modify experiments, and apply various evaluation procedures through sets of modular graphical interface including pop-up menus, dialogue boxes, spread-sheet like windows, and plotting windows. The activation of each tool is selected from a pull-down menu. Simulation process of each tool is carefully monitored and the error handler module in the software shell will process system errors and provide the user with appropriate error message.

For a coupled transmission line circuit simulation, alternative sets of models can be compared and ranked through a scheme called the experimental frame [7,8]. An experimental frame contains sets of conditions under which a design model is simulated. PDSE allows up to ten variations of package models, and ten different experimantal frames to be simulated at the same time. Each package model is a coupled transmission line system including terminating networks and coupled conductors' cross sectional geometries. The experimental frame is defined independently of the models. It consists of a set of input waveforms (for example, step inputs with different rise times or generic waveforms defined by piecewise linear segments) 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 (in this case, experimental frame specifications). 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.

To efficiently construct the interconnect geometries for parasitic extraction, PDSE also provides translators for importing external database. For example, DXF (Drawing eXchange Format) translators are provided for package designers using AutoCAD as their layout tool. In this manner, the already created interconnect geometries can directly be interfaced with 2D or 3D electrical parameter tools in PDSE for parasitic extractions [6]. The other feature of PDSE includes an embedded expert system for reducing signal delay and crosstalk by changing the cross sectional geometry of a given microstrip line structure [9].

The tightly-coupled scheme for tool integration as described in this layer of framework is straightforward since each tool uses its own data model as input data structures. However this might cause inconvenience for end-users as the number of tools integrated in the framework increases. For the case that several tools are invloved for an analysis, users have to generate different data structures conformed to individual tool's requirement or even the interactions among them. This is due to the lack of uniform data representation originated from the nature at the early stage of tool development as explained in Section 1 of this paper. A loosely-coupled scheme which provides a uniform representation of design data to support different simulations is suggested and implemented in the layer II of our framework. A common database which consists of libraries for simulation model genertions is established. In this manner, the applications (simulations) cluster around a smaller set of design data objects even if new tools are going to be developed and integrated in the future.

3. Design data representation and management

The complexity of electronic packaging/interconnect structures has increased the difficulty of managing design data for computer-aided design. With more packaging simulators developed for analysis and design verification, effective representation and management of design data have become a critical issue. To support various tools for simulations, we believe a design database should be designed in such a way that both electrical and thermal/mechanical applications can access a common database in a unified fashion. In other words, the framework should provide a unified data model with views to provide access to appropriate projections. This will provide the benefits of preserving data entities and relationships as applications move. Moreover, it will facilitate incremental change and alleviate the load of both CAD tool integrators and end-users for managing design data for simulations. To reduce the complexity of a design, the use of hierarchy and abstraction is also essential. In this level of framework, two major libraries called the Chip Model Library (CML) and the Package Model Library (PML) are established and served as a common database to support modeling and simulation of the first level (chip level) and the second level (board level) packages. The CML includes physical, thermal and electrical (especially peripheral circuit models of a chip such as output buffers and receivers) information of chips with different levels of complexity. The PML consists of multilevel off-chip interconnect structures (discontinuities such as bends, vias, and crossovers, etc. are also included). Generation of models for simulation requires effective interface and coupling between these two major libraries. As shown in Fig. 3, the underlying architecture of this level of framework consists of three major components: design database, CAD tool integration and simulation management. The design database (i.e., the CML and the PML) and CAD tool integrations are realized in an object-oriented approach using C++ programming language [10,11]. Each CAD tool is encapsulated as an object in the CAD tool integration layer. In each tool object, several methods are usually provided as translators. Note that a method here means a routine performing an operation on an object [10]. The performance of file conversions is twofold (as shown in Fig. 3). 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 other CAD tools. All these details are encapsulated in a software object and transparent to the users. This relieves the users from intensive interactions with the details of the packaging simulators. The third layer of the architecture is for post processings of

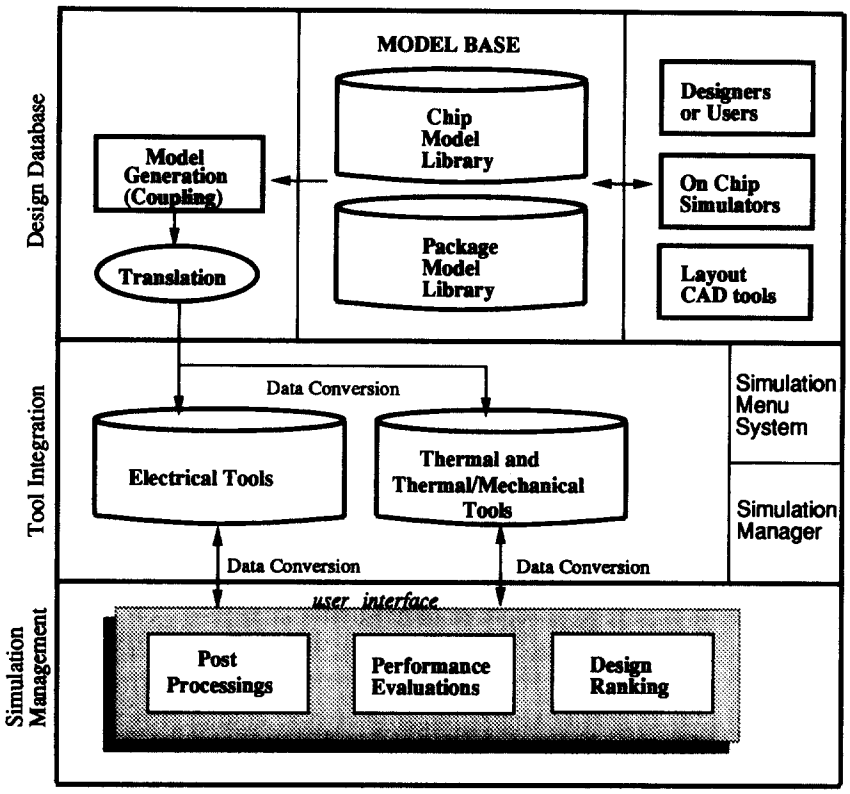


Fig. 3. The second level of the framework: design data representation and management.

the simulation results such as graphical displays or error handlings. Performance evaluation of the simulated package structure is also proceeded in this layer.

3.1. The Chip Model Library

As illustrated in Fig. 4, the Chip Model Library(CML) is organized as a hierarchical structure [12].

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 different structures and applications required. A NAND gate, for example, can be further

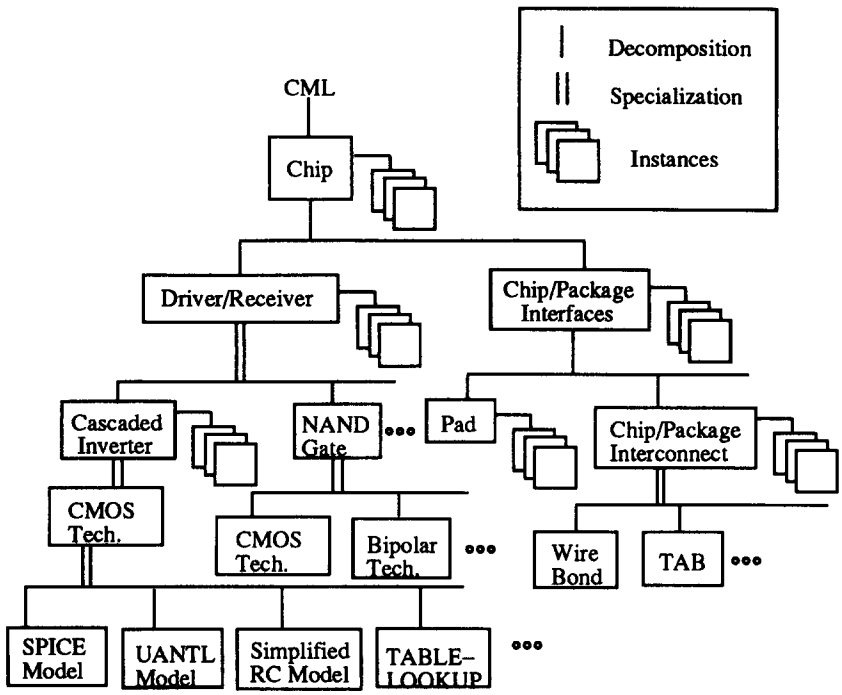


Fig. 4. A structural representation of the Chip Model Library.

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 CML 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.

Figure 5 depicts C++ code fragments to illustrate how the object-oriented approach can be applied to construct a UANTL CMOS cascaded inverter from PMOS NMOS objects. The UANTL [13] is a lossless transmission line simulator developed at the University of Arizona with SPICE like network specifications.

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 [14]. 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 simulations can operate with the same hierarchy. A typical chip model frame is shown in Fig. 6.

```
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;}
};
```

Fig. 5. The implementation of the CML using an object-oriented approach.

3.2. The Package Model Library

The Packaging Model Library [15,16] encompasses models for different types of boards, modules (package substrate) and multilevel interconnects. The structural representation of the PML is shown in Fig. 7. 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 interconnects among and outside of chips. Package pins, wire bonds, vias between

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

Fig. 6. A frame based representation of a chip model.

two wiring levels, conductor bends, and line width variations on a single level can be modeled as individual inductive and capacitive discontinuity. Since discontinuity will cause signal reflection, they must be considered in a design to minimize the resulting effects. Currently discontinuities in the PML includes models of bends, vias and tapered lines. To facilitate circuit simulation, these discontinuities are usually modeled as equivalent lumped circuits with capacitive and inductive elements. In the PML, loop-up tables [17] are created which contain numerical values of capacitive and inductive elements of typical structures of bends and vias. Conductors can be modeled as transmission line or a lumped capacitor. It is dependent on the signal rise time and transmission line time-of-flight delay, if the former is less than or comparable to the latter, transmission line behavior becomes significant. Transmission line can be lossy or without loss. The cross section of a conductor can be of different geometry, typically rectangular or trapezoidal, etc. The board and module wires are sufficiently wide and thick to be treated as lossless transmission lines. On-chip interconnections and some thin film package wires, however, have significant resistance, and they should be treated as lossy transmission lines. For the signal line construction, it can be single layer substrate with two reference planes to sandwich the conductors (stripline structure) or multilayered substrates with one reference plane (microstrip structure).

Fragments of C++ codes which are used to illustrate how we developed a transmission line system in the PML are shown in Fig. 8. Note that in this example, if a lossless transmission line model is considered as a generalization with respect to a lossy transmission line model which in this case is a specialization, then one can use the property of inheritance provided by object-oriented programming

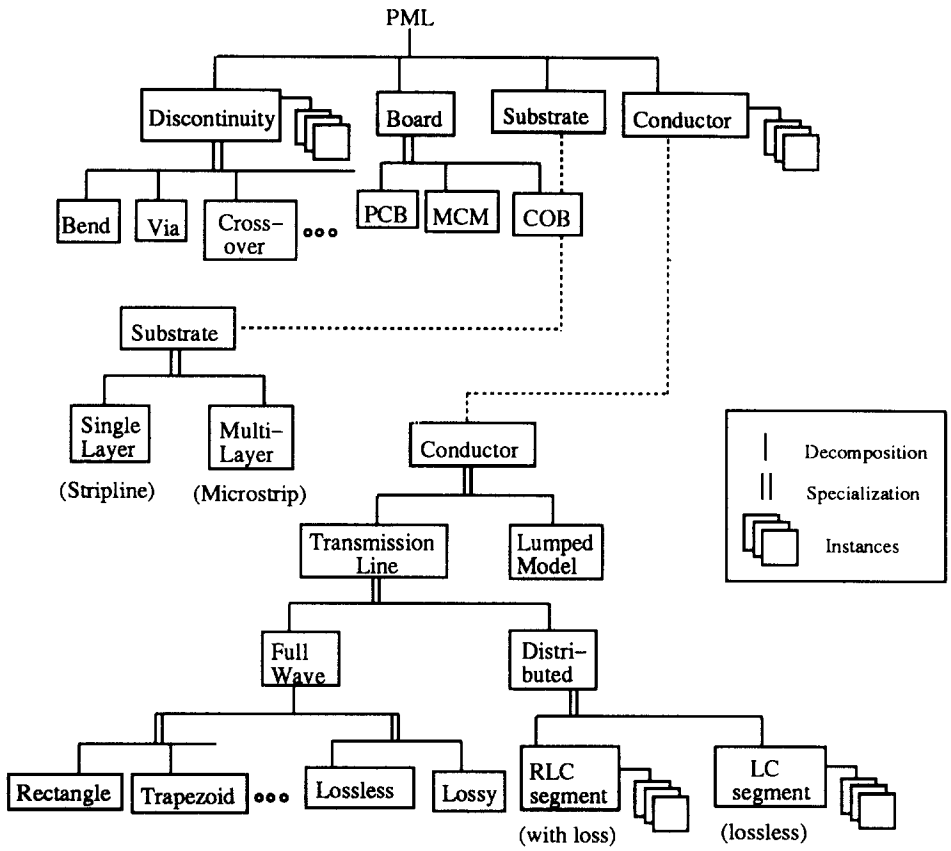


Fig. 7. A structural representation of the Package Model Library.

language to derive a new class for the lossy transmission line model from its base class (in this case, the class for lossless transmission line model).

3.3. Simulation model generation

Simulation model generation is achieved by specifying the interconnections or couplings between the PML and the CML. Currently, this level of framework provides two types of coupling schemes: (1) the chips and packaging interconnects (e.g., chips on a printed circuit board) and (2) the chip to chip interconnects (e.g., an off-chip interconnects with termination circuitry on the chips). The coupling and retrieval of models from the PML and the CML are achieved from constructing a high level task specification called the packaging description file (PDF) [16]. A typical PDF is shown in Fig. 9. PDF is specified to flatten the hierarchical structures by specifying the interconnections in the PML and the CML so that simulation algorithms can be applied. Each simulation tool integrated in our framework has its own translators encapsulated to transform related parts in the PDF to the corresponding input file for simulation. In the case of Fig. 9, three chips are placed on a printed circuit board and a transmission line system is specified between chip A (the driving chip) and chip

```
class lossless_rec_TLsys // lossless transmission line system
{
  protected:
     char myname[20]; // model name
     int Ncndr; // total number of conductors
     float Lcndr; // conductor length, unit:cm
     float s; // spacing
     float h; // height to the ground plane
     float w; // conductor width
     float t; // conductor thickness
  public:
     lossless_rec_TLsys(); // constructor
     void define_node(); // define connection node
     // window based input menu
     virtual void input_cross_section(Window,int,int);
     virtual void get_name(Window); // a dialog box for model name
     virtual int save_TLsys(); // save a model
};
     // define a lossy transmission line system
     // use lossless transmission line system as a base class
class lossy_rec_TLsys : public lossless_rec_TLsys
{
  public:
     float sig; // conductivity of the conductor (siemens/m)
     float mu; // magnetic permeability of the conductor
     void input_cross_section(Window,int,int);
     void get_name(Window); // a dialog box for model name
     int save_TLsys(); // save a model
};
```

Fig. 8. The implementation of the PML using an object-oriented approach.

B (the receiving chip). More detailed examples for the use of the PDF for simulation will be given in Section 5.

4. System level modeling and simulation

In the first two layers of the framework as discussed in Sections 2 and 3, simulations are performed using only detailed tools (i.e., rigorous electromagnetic solvers). For the early system design of multichip packages such as multichip modules (MCMs), usually it is complicated by many trade-offs resulting from conflicting design constraints and parameters. To globally optimize all performance matrices such as module clock frequency, noise, power dissipation, etc., a designer has to consider numbers of design parameters, their interdependencies and trade-offs simultaneously [18]. For example, higher system clock frequency usually requires higher chips integration and proper interconnect/package technologies. However, higher module frequency and chip integration increase the power dissipation. Due to the noise to signal ratio, crosstalk and simultaneous switching noise might limit the level of chip integration and interconnect technology used, thereby higher module frequency

```
BOARD:
  begin
    myPCB
  end
CHIPS:
  begin
    chipA chipB chipC
  end
SUBSTRATES:
  begin
    2Layer
  end
TRANSMISSION_LINE_SYSTEM:
  begin
    DRIVERS:
      begin
      chipA.pad1.inv(UANTL) connection(10) signal: my0.5n (1,0)
      chipA.pad2.inv(UANTL) connection(20) signal: myDC5v (2,0)
      end
    RECEIVERS:
      begin
      chipB.pad1.inv(UANTL) connection(11)
      chipB.pad2.inv(UANTL) connection(21)
      end
INTERCONNECTS:
    begin
      my2line
    end
  end
```

Fig. 9. A fragment of PDF for a transmission line simulation.

might not be achieved for a given MCM design. To make early design trade-offs before physical implementation of a multichip system such as an MCM, it is essential to be able to estimate system performance and analyze the interdependencies among design parameters. A direct approach is to apply detailed tools such as those provided in the first and second layers of the framework for the assessment of electrical and thermal peformance characteristics of a pre-selected candidate MCM. However, it is usually computationally expensive. Besides, it is difficult to predict the dependencies among design parameters and performance trends by just performing one or two simulations. The other approach is to apply closed-form equations for the estimation of performance matrices. A module or board level package is modeled from a macro-scopic point of view and thus rapid design evaluation can be achieved. This approach might not be numerically accurate (acceptable accuracy is desirable, however), but the overall picture of the system performance can be easily captured by this scheme. The system designers can investigate the trade-offs and interdependencies among design parameters more efficiently. In the third layer (the highest abstraction level) of our framework, we adopted a trade-off approach for the early assessment of an MCM design: closed-form equations are used for global analysis; detailed simulations tools are used for local analysis [19] . The estimations of system performance and interdependencies among various design parameters obtained from the

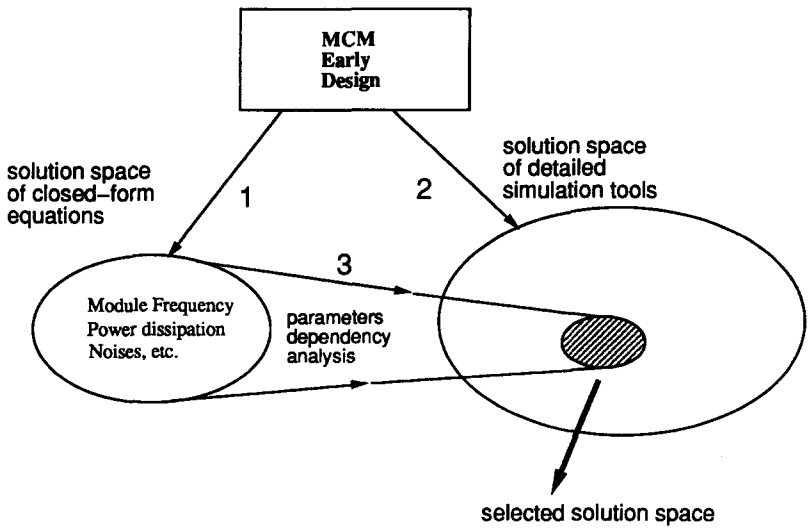


Fig. 10. Three different approaches for the early design of an MCM.

solution space of closed-form equations are used to guide the system designer in selecting appropriate simulations for further design verifications using detailed tools provided in the level I or the level II of our framework. The idea is illustrated as approach 3 in Fig. 10.

In this layer of framework, a system level simulation environment is developed to facilitate the early design of multichip systems especially for the CMOS based implementation of MCMs. Electrical system performance matrices such as module clock frequency, etc. are predicted and analyzed for different interconnect and packaging technologies. Performance-limiting factors such as power dissipation, coupled noise, and simultaneous switching noise are also included in this layer. Design parameters may vary depending on the purposes and technological limitations of the MCM vendors. In this framework, the following parameters are included for study:

- *Interconnect technology*: This includes interconnect cross section geometry, thin film or thick film technology, stripline or microstrip line structure, substrate dielectric, conductor resistivity, etc.
- *Logic technology*: The characteristics associated with the CMOS output driver device and receivers, the supply voltage, etc.
- *Chip technology*: Number of logic chips, memory chips. The average chip size ratio between logic chip and memory chip. Chip's footprint size. Number of I/O's for each chip.
- *Bonding technology*: Flip chip, wire bonding, or TAB (tape automated bonding).
- *Effective inductance*: Chip–package inductance (for example, a bond wire inductance) for a single connection, V_{SS} plane inductance, inductance for via connections, and package–pin inductance for a single connection.

Based on design (technology) parameters just mentioned, closed-form equations can be derived for the estimation of system performance matrices including module clock frequency, module size, power dissipation and simultaneous switching noise. The detailed derivation and application limitations of the closed-form equations for module clock frequency approximation are included in the appendix of this paper. The rest of the closed form equations can be found in [19].

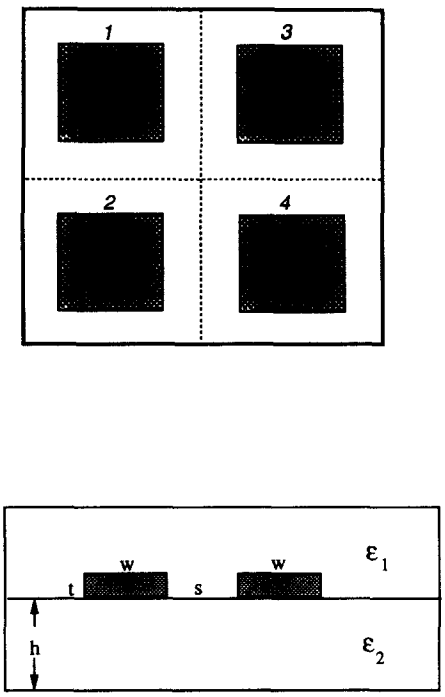


Fig. 11. A four chip MCM and its interconnects.

5. Examples

In this section, we use examples to illustrate how the three layered framework mentioned so far can be used to facilitate the modeling and simulation for electronic packages.

A four chip, thin film MCM as shown in Fig. 11 is used for our case study. The design parameters provided by an MCM supplier is classified as follows:

- *Interconnect technology*: The thin film microstrip line structure is used for interconnects. The dimensions of the cross section geometry of the conductors selected for study are: $w = 10\,\mu$m, $t = 2\,\mu$m, $h = 10\,\mu$m, $s = 10\,\mu$m (as shown in Fig. 11 also). Copper is selected as conductor material with resistivity $\rho = 1.7\,\mu\Omega \cdot$cm. Relative permittivity ε_1 for the microstrip line structure is equal to the permittivity of free space. We vary the values of ε_2 (the relative permittivity of the MCM substrate) so that the impact of different substrate dielectric on the system performance can be studied. Eight different materials are selected for our simulations: Alumina (ε_r=9.5), Beryllia (ε_r=6.7), Epoxy glass (ε_r=5.0), Polyimide (ε_r=2.5), Silicon dioxide (ε_r=3.9), Silicon nitride (ε_r=7.5), Gallium arsenide (ε_r=10.9), Silicon (ε_r=11.7).
- *Logic technology*: The values of ON-resistance R_{tr} and input capacitance C_{tr} of a minimum-sized nMOS/pMOS transistors (smallest pre-driver of the cascaded drivers) are 300 Ω and 15 fF respectively. The input gate capacitance (C_{rec}) of a receiver is 0.5 pF. The supply voltage for CMOS devices is $V_{DD} = 5.0$ V.
- *Chip technology*: The total number of equivalent chips N_c integrated in the MCM is four (we assume four equal sized chips in this case study). It is assumed that the clock frequency of

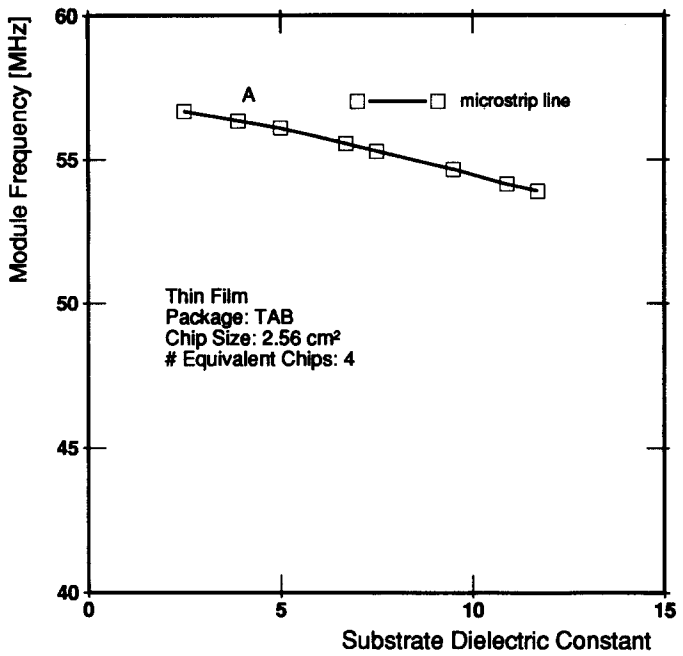


Fig. 12. The interdependency between module frequency and substrate's dielectric constant.

a single driving chip (f_c) is 60 MHz. The worst case fanout f_o considered in this analysis is equal to ($N_c - 1$). The chip's footprint size is 2.56 cm^2 with 180 I/O's.

- *Bonding technology*: The TAB (Tape-Automated Bonding) is selected as chip to package interconnection. The equivalent capacitance value of a TAB bonding is C_{pad} = 1.5 pF.

In the following three sections, simulations using the functionalities provided in three different layers of the framework are performed. We first apply the closed form equations in layer III to estimate the system performance (module clock frequency) based on different dielectric materials. We then localize one of the MCM system configurations and perform detailed simulation using the rigorous tools in layer I of the framework. It is also assumed that chip and package models for the MCM studied in this case study are already available in the CML and the PML. Thus, similar simulations using rigorous tools can be performed in the layer II of the framework by coupling the related models.

5.1. Simulation in layer III

By applying the closed form equations in layer III, we can calculate the module clock frequency based on the design parameters provided in the problem description. Since we vary the substrate's dielectric constant for a given MCM's system configuration (including interconnects, logic, chip set, bonding, and effective inductance), the interdependency between substrate's dielectric constant and module clock frequency can be estimated and analyzed as shown in Fig. 12.

Note that, if we prefer to have a system's clock frequency around 56 MHz, then three substrate materials can meet the requirement: Polyimide (ε_r=2.5), Silicon dioxide (ε_r=3.9) and Epoxy glass (ε_r=5.0). Although all the three materials are feasible for our application, Silicon dioxide is selected

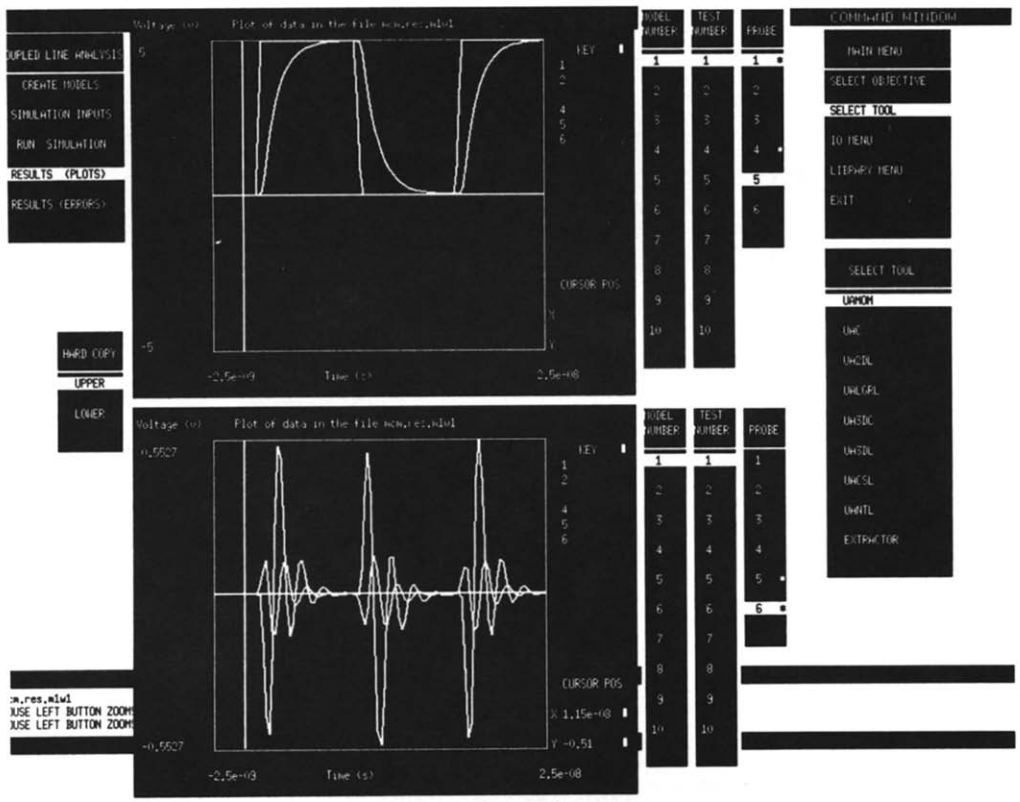


Fig. 13. The transient response of the MCM equivalent circuit.

due to its minimal cost. This MCM is depicted as point A in Fig. 12. We will perform detailed simulations based on the configuration represented by point A in next section.

5.2. Simulation in layer I

To map the point represented by point A in Fig. 12, we have to use the equivalent circuit described in the appendix. Notice that the circuit in Fig. 20 is exactly applicable to our case study since the driving chip has three fanouts. For driving chip with more fanouts, additional capacitive receiving branches have to be added in the equivalent circuit for detailed simulation. The values of R_{dout} and R_{int} are the simulation results from layer III of our framework. For this case study, $R_{dout} = 20\,\Omega$ and $R_{int} = 54.4\,\Omega$. To represent a 60 MHz driving chip, we construct an input waveform with rise time (T_r) and fall time (T_f) both equal to 0.5 ns. The transient response of the equivalent circuit simulated using UANTL is shown in Fig. 13.

5.3. Simulation in layer II

This section describes simulations using layer II of the framework. Suppose we have chip models for the four chips on the considered MCM as shown in Fig. 11. The frame based representataion for chip 1 of the MCM is used as an example and shown in Fig. 14.

```
ChipName: chip1
 Thermal Slot:
    Rja: not specified
    Rjh: not specified
    Pd: 20000
    Sp: not specified
 Physical Slot:
    dx: 0.0096
    dy: 0.0096
 Electrical Slot:
    CMOS
    PAD # 1
    Cascaded Inverter (UANTL)
    stage: 2
    ratio: 3
    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
    PAD # 2
    Cascaded Inverter (UANTL)
    stage: 1
    ratio: 1
    pMOS parameters
    .....
```

Fig. 14. The frame based representation of chip 1 in the MCM.

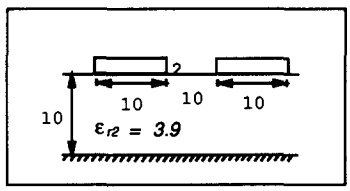
The detailed interconnect model in the PML is shown in Fig. 15. To specify the the chip placement on the MCM substrate and the off-chip interconnect between chip 1 and chip 3, we need to provide a package description file as described in Section 3 to retrieve and couple models from the CML and the PML for simulations. The package description file for this case study is shown in Fig. 16.

The package description file is interfaced with both UANTL and UATPCB [20], therefore both electrical and thermal design verifications can be accomplished. As shown in Fig. 17, transient response of the transmission line system between chip 1 and chip 3 is simulated. The temperature rise and contours for the four chip MCM are shown in Fig. 18 and Fig. 19 respectively.

6. Related work

Not many CAD frameworks which incorporate three levels of services (functionalities) as described in this paper for modeling and simulation of electronic packaging and interconnects can be found in literature.

For the scheme of design data representation and management in the second layer of our framework, a similar semantic data model for VLSI design and their relationships in a unified way is presented in [21]. The relationships among data entities in [21] include IS-OF-TYPE, IS-A-KIND-OF, IS-COMPOSED-OF and MANY-TO-MANY which are similar to the concepts of classes/objects (instances), specialization, decomposition described in this paper. To provide a "flattened" net list data from a hierarchical database for simulations, a library of functions called hierarchy walk manager



Interconnect and substrates objects

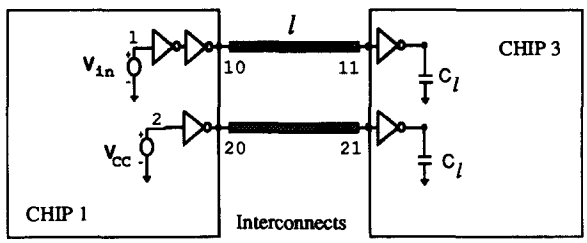


Fig. 15. Simulation of a transmission line system using the CML and the PML.

```
BOARD:
 begin
  myMCM
 end

CHIPS:
 begin
  chip1 chip2 chip3 chip 4
 end

SUBSTRATES:
 begin
  myMICROSTRIP
 end

TRANSMISSION_LINE_SYSTEM:
 begin
  DRIVERS:
    begin
     chip1.pad1.inv(UANTL) connection (10) signal: my0.5n (1,0)
     chip1.pad2.inv(UANTL) connection (20) signal: myDC5V (2,0)
    end
  RECEIVERS:
    begin
     chip3.pad1.inv(UANTL) connection(11)
     chip3.pad2.inv(UANTL) connection(21)
    end
  INTERCONNECTS:
    begin
      my2LINE
    end
 end
```

Fig. 16. The package description file of a four chip MCM.

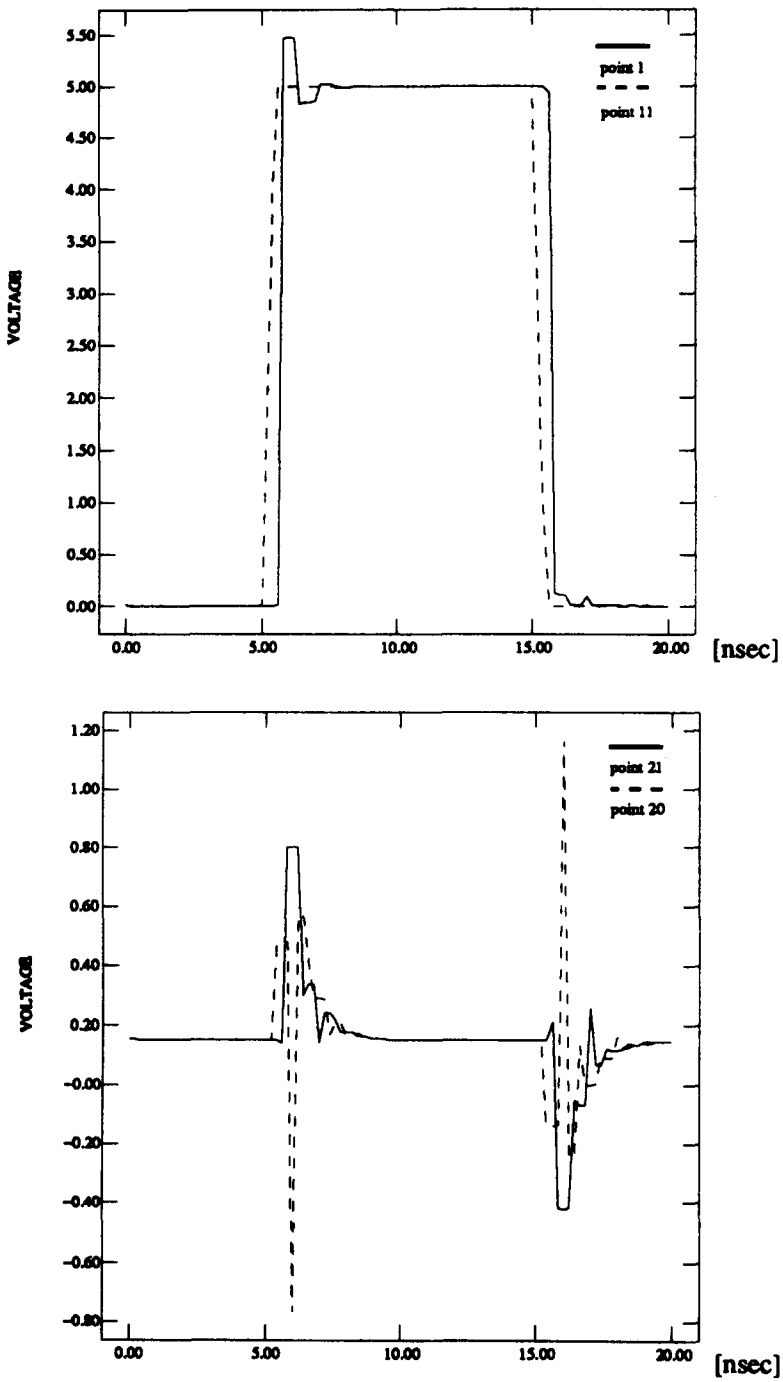


Fig. 17. The transient response of the transmission line system defined between chip 1 and chip 3.

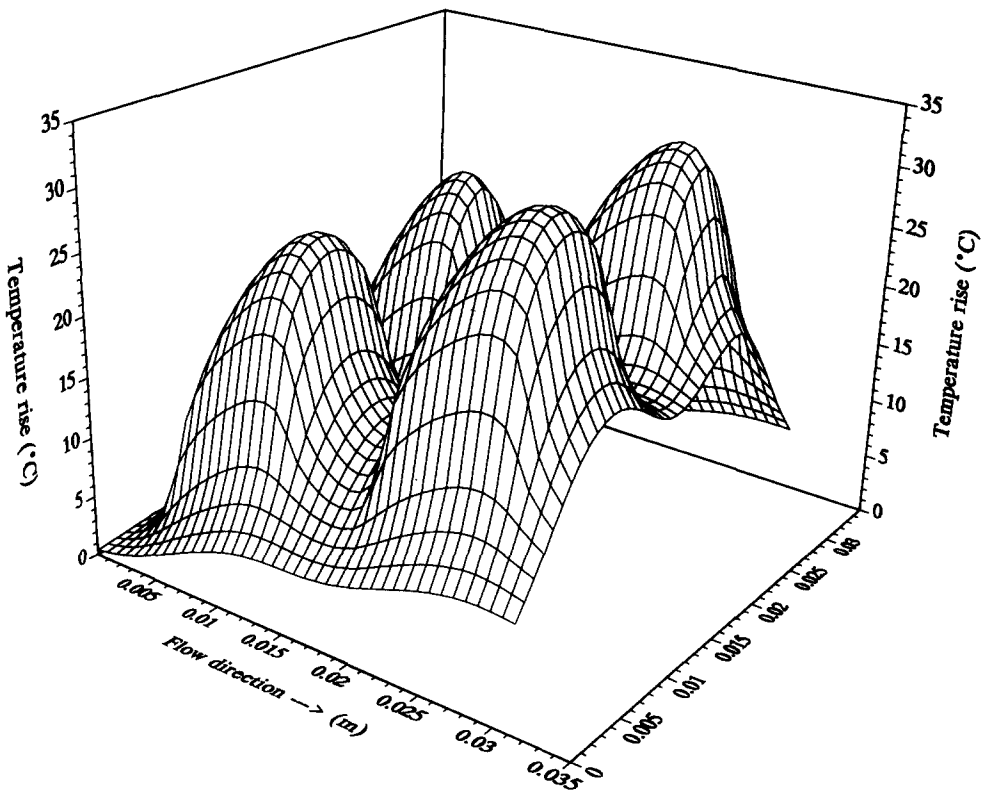


Fig. 18. Temperature rise due to four chips on an MCM.

is built to traverse the design hierarchy in a convenient way.

SLIP [22] is a framework for algorithms that partition and implement complex VLSI-based systems as a hierarchy of electronic packages. SLIP provides a policy for representing hierarchical systems on the OCT data model as well as an attribute mechanism which is used to annotate the representation and to build models of the systems.

FACE (Flexible Architecture Compilation Environment) [23] provides a common object-oriented representation shared among CAD tools. It supports hierarchical design, multiple views of the design data, and provides a common data access mechanism. The concept discussed in [23] is the most similar to our work although the applications are different in many respects. FACE is written in Lisp with object-oriented extensions.

STAIC [24] is an interactive framework for synthesizing CMOS and BiCMOS analog circuits. Circuit descriptions in STAIC can be classified as being of categoric, specific, device, or layout type in a hierarchical fashion which is similar to how we organize the PML and the CML. STAIC uses C++ object-oriented programming language to construct fundamental syntax of an input modeling language for entering hierarchical circuit descriptions.

For system level modeling and simulation, although other tools have different aspects of functionalities, very few of them consider noise as a performance-limiting factor and are integrated with detailed simulation environment for further analysis which is essential for practical MCM early designs.

Developed at Stanford University, SUSPENS [25] is basically an analytical model that takes

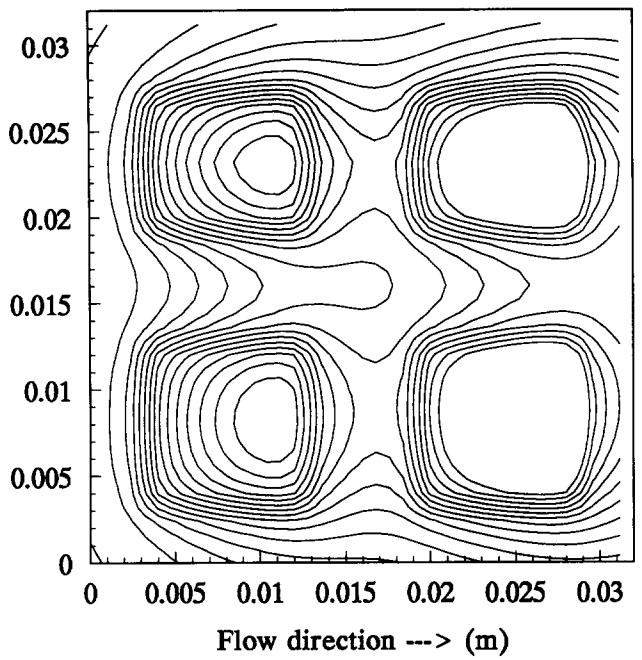


Fig. 19. Temperature rise contour for four chips on an MCM.

some material, device, circuit, logic, package, and architecture parameters as inputs and using them calculates the clock frequency, module size and power dissipation of the system, as well as several other detailed parameters. The results of this work have been compared with existing microprocessors, and used to predict the performance of future microprocessors. Some of the equations for calculating the module frequency have been modified and included in our simulation environment.

AUDiT [26] is a package system simulation software used to study multi-chip module trade-offs. Modules in AUDiT are described with a set of structural, electrical and materials related model parameters. Simulated annealing technique is applied to determine the structure and technologies required for a optimal (minimum cycle time) system. Three packaging hierarchies (single chip, module, and board) based on CMOS technology have been used as demonstration systems. Logic and memory subsystems were treated separately in this work.

SPEC [27] is designed to compute geometric, electrical, thermal, and manufacturing(cost) performance metrics specifically for multichip systems. The estimations used in SPEC "are based on those previously developed in recent literature" [27]. For electrical performance, the system did not include noise as a performance-limiting factor. The system also lacks more detailed simulation tools to analyze the effects of signal degradation and reflection which cannot be estimated or predicted from first order closed-form equations.

7. Conclusion

In this paper, an integrated framework to support modeling and simulation of VLSI packaging and interconnects has been presented. Based on the software tools developed at the University of

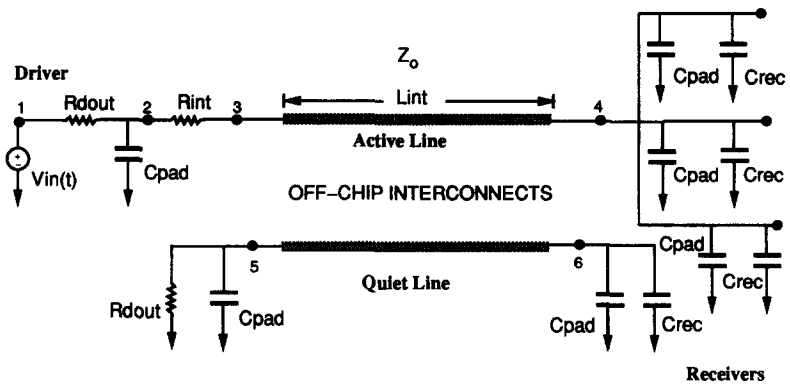


Fig. 20. An equivalent MCM circuit model for analysis.

Arizona, we introduced a new scheme to manage design data and CAD tools involved for modeling and simulations of electronic packaging. A three layered framework architecture has been proposed and implemented to provide different levels of services and abstractions. An object-oriented approach has been applied to develop libraries and encapsulate CAD tools. Simulation model generation through a centralized database has been demonstrated. System (board or module) level modeling and simulations of electronic packaging are described and illustrated.

Appendix

Closed form equations used to estimate the module frequency and module size of an MCM are included in this section. These equations are part of the core formulas applied for the implementation of level III of the described framework.

The equivalent MCM circuit for calculating the system clock frequency is shown in Fig. 20, where Rd_{out} (Ω) is output driver resistance, C_{pad} (pF) is bond pad capacitance, $V_{in}(t)$ (Volt) is the driving voltage source, R_{int} (Ω/cm) is per unit length (p.u.l.) interconnect resistance, C_{int} (pF/cm) is p.u.l. interconnect capacitance, C_{rec} (pF) is the input capacitance of a receiver, and f_o is the fanout of the output driver(for example, $f_o = 3$ in Fig. 20). R_{int} is included before the off-chip interconnect for the consideration of losses, although in detailed model, this value is distributed along the interconnect.

In this work, it is assumed that the width and length of the MCM are of equal dimension. For the calculation of interconnect length L_{int} (cm), the worst case chip-to-chip delay path is assumed by calculating two times of the dimension of the module:

$$L_{int} = 2\sqrt{F_p N_c}, \tag{1}$$

where F_p (cm^2) is the footprint size of a logic chip, which is limited by module wiring capacity and bond pad, and N_c is the total number of "equivalent chips" in an MCM. Usually in a multichip module, there are logic chips and memory chips. And the size (chip area) of memory chip is different from logic chip. The number of "equivalent chips" is here defined as the number of "equivalent logic chips", and this term will be used for the rest of the discussions.

If P is the total number of logic chips, Q is the total number of memory chips in a multichip module, and ζ is average chip size (area) ratio between logic and memory chips integrated onto the module, then the number of equivalent chips (N_c) is

$$N_c = (P + \frac{Q}{\zeta}). \tag{2}$$

And the module size M_{size} (cm^2) can then be estimated as

$$M_{size} = N_c F_p. \tag{3}$$

Initial work on module frequency estimation for systems is given by Bakoglu [1]. In this work, some of the equations are modified especially for CMOS based systems. These equations are listed and described below.

Cascaded drivers (inverter chains) are applied in the system model because of the consideration of large fanout in MCM application. The driving-chip (on-chip) delay T_c can be calculated from

$$T_c = \frac{1}{f_c} + 2e(N-1)R_{tr}C_{tr} \times 10^{-15}\,\text{s}, \tag{4}$$

where f_c is chip operating frequency in an MCM, and e (e=2.71...) is the optimal ratio of two consecutive drivers. R_{tr} (Ω) and C_{tr} (fF) are the ON-resistance and input capacitance of a minimum-sized nMOS and pMOS transistors, (i.e. R_{tr}(nMOS) is equal to R_{tr}(pMOS)). The number of stages N can be calculated by setting the ON-resistance of the output stage to Z_0 (off-chip interconnect characteristic impedance) and pessimistically assuming that the first inverter of the cascaded chain is a minimum-sized device:

$$N = 1 + \ln(\frac{R_{tr}f_0}{Z_0}). \tag{5}$$

In addition to chip delay T_c, RC delays (T_{m1} to T_{m5}, as shown below) which consider the charging/discharging of the driver, receivers and interconnect capacitance, and transmission line propagation delay (T_{m6}) need also to be included for the module delay calculation.

$$T_{m1} = (1 + f_0)(C_{pad}R_{dout}) \times 10^{-12}\,\text{s}, \tag{6}$$

$$T_{m2} = \frac{1}{2}(R_{int}L_{int})(C_{int}L_{int}) \times 10^{-12}\,\text{s}, \tag{7}$$

$$T_{m3} = R_{dout}(C_{int}L_{int})f_0 \times 10^{-12}\,\text{s}, \tag{8}$$

$$T_{m4} = R_{dout}C_{rec}f_0 \times 10^{-12}\,\text{s}, \tag{9}$$

$$T_{m5} = R_{int}L_{int}C_{rec} \times 10^{-12}\,\text{s}, \tag{10}$$

$$T_{m6} = \frac{L_{int}}{V_m} \tag{11}$$

where V_m is the propagation velocity. Notice that these equations also include bond pad parasitics and consider the effect of fanouts for CMOS implementation. The module frequency M_f(MHz) can then be calculated as

$$M_f = (T_c + \sum_{i=1}^{6} T_{mi})^{-1}. \tag{12}$$

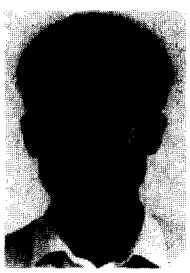
Since our simple model forms a RC tree like network, to determine the delay with higher accuracy on a specific branch, one needs to calculate the contribution of each other branch on the branch of interest for this RC tree. Detailed analysis on the calculation of delay using RC tree approach is given in [28]. In this application, calculated interconnect length L_{int} is adopted only when the error of off-chip delay is within 10% when compared with that determined by detailed tools.

Also note that, with stub-like loading, bends and vias, our simple approach will give more than 25% error for long interconnect length, and more detailed simulations are essential to predict the maximum module frequency.

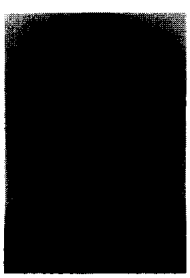
References

[1] H.B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI* (Addison-Wesley, Reading, Mass., 1990).
[2] 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.
[3] J.E. Morris, *Electronic Packaging Forum, Vol. 1* (Van Nostrand Reinhold, New York, 1990) Ch. 4.
[4] D.S. Harrison, A.R. Newton, R.L. Spickelmier and T.J. Barnes, Electronic CAD Frameworks, *Proceedings of the IEEE*, Vol. 78, No. 2, pp. 393–417, 1990.
[5] J.W. Rozenblit, J.L. Prince, O.A. Palusinski and T.D. Whipple, Computer Aided Design System for VLSI Interconnects, *Proc. IEEE Intl. Conf. on Computer Design: VLSI in Computer and Processors*, pp. 237–241, 1989.
[6] P. Hsu, S. Voranantakul, J.W. Rozenblit and J.L. Prince, Intelligent Design Automation of VLSI Interconnects, *Proc. IEEE Intl. Phoenix Conf. on Computers and Communications*, pp. 349–355, April 1992.
[7] J.W. Rozenblit and B.P. Zeigler, Design and Modeling Concepts, in: *International Encyclopedia of Robotics Applications and Automation* (Wiley, New York, 1988).
[8] B.P. Zeigler, *Object-Oriented Simulation with Hierarchical Modular Models: Intelligent Agents and Endomorphic Systems* (Academic Press, New York, 1990).
[9] T. Simunic and J.W. Rozenblit, Reduction of Signal Delay and Crosstalk in Electronic Packages Using Expert System Techniques, *Proc. 2nd IEEE Topical Meeting on Electrical Performance of Electronic Packaging*, October 20–22, 1993, pp. 61–64.
[10] B. Stroustrup, *The C++ Programming Language* (Addison-Wesley, New York, 1991) Ch. 12.
[11] I.Jacobson, *Object-oriented Software Engineering* (Addison-Wesley, New York, 1992).
[12] S.N. Pratapneni, C.M. Wolff, P. Hsu, J.W. Rozenblit and J.L. Prince, Development of Chip Model Library for the Computer-Aided Analysis of Electronic Packaging, *Proc. IEEE/CHMT International Electronic Manufacturing Technology Symposium*, Santa Clara, Calif., October 4–6, 1993, pp. 417–422.
[13] 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.
[14] R. Fikes and T. Kehler, The Role of Frame-Based Representation in Reasoning, *Communications of the ACM*, Vol. 28, No. 9, pp. 904–920, 1985.
[15] P. Hsu, J.W. Rozenblit and J. Prince, An Object-Oriented Approach for the Hierarchical Design of Multichip Systems, *Proc. 10th International VLSI Multilevel Interconnection Conference*, Santa Clara, Calif., June 7–10, 1993, pp. 538–540.
[16] P. Hsu, J.W. Rozenblit, S.N. Pratapneni and C.M. Wolff, An Integrated System for Design Automation of VLSI Interconnects and Packaging, *Proc. 36th IEEE Midwest Symposium on Circuits and Systems Conference*, Detroit, Mich., August 16–18, 1993, pp. 847–850.
[17] Srinivasan Venkataraman, Private Communications, University of Arizona, 1993.
[18] P. Hsu, R. Senthinathan, J.W. Rozenblit and J.L. Prince, Electrical Performance Evaluation for Multichip Assemblies Using Knowledge-based Approach, *Proc. 1st IEEE Topical Meeting on Electrical Performance of Electronic Packaging*, 1992, pp. 7–9.

[19] P. Hsu, R. Senthinathan, J.W. Rozenblit and J.L. Prince, A Knowledge-based Simulation Environment for the Early Design of Multichip Modules, *Microelectronics Journal*, Vol. 25, No. 5, p. 47, 1994.
[20] A. Ortega, S. Ramanathan and J. Prince, A Point Source Thermal Wake Model For Heated Components in Turbulent Channel Flow, *Proc. 1992 Intl. Electronic Packaging Conference*, Austin, Texas, September 27–30, 1992, pp. 532-547.
[21] A. Singhal, N. Parikh, D. Dutt and C. Lo, A Data and Architecture for VLSI/CAD Databases, *Proc. ICCAD*, 1989, pp. 276-279.
[22] M. Beardslee, C. Kring, R. Murgai, H. Savoj, R.K. Brayton and A.R. Newton, SLIP: A Software Environment for System Level Interactive Partitioning, *International Conference on Computer-Aided Design*, 1989, pp. 280-283.
[23] W.D. Smith, J.R. Jasica, M.J. Hartman and M.A. d'Abreu, Flexible Module Generation In The FACE Design Environment, *International Conference on Computer-Aided Design*, 1988, pp. 396-399.
[24] J.P. Harvey, M.I. Elmasry and B. Leung, STAIC: An Interactive Framework for Synthesizing CMOS and BiCMOS Analog Circuits, *IEEE Trans. on Computer-Aided Design*, Vol. 11, No. 11, p. 1402, 1992.
[25] H.B. Bakoglu and J.D. Meindl, A System Level Circuit Model for Multi- and Single-chip CPUs, *IEEE International Solid State Circuits Conference*, New York, February 1987, pp. 308–309.
[26] J.P. Krusius, System Interconnection of High Density Multi-Chip Modules, *SPIE Vol. 1390 International Conference on Advances in Interconnection and Packaging*, 1990, pp. 261–270.
[27] P.A. Sandborn, A Software Tool for Technology Tradeoff Evaluation in Multichip Packaging, *Proc. 1991 IEMT Symposium*, September 1991, pp. 337–341.
[28] J. Rubinstein, P. Penfield, Jr. and M.A. Horowitz, Signal Delay in RC Tree Networks, *IEEE Trans. on Computer-Aided Design*, Vol. CAD-2, No. 3, pp. 202–211, 1983.



Pochang Hsu received the B.S. and M.S. degrees from the National Cheng-Kung University, Taiwan, Republic of China in 1984 and 1986, respectively, and the Ph.D. degree from the University of Arizona, Tucson, USA in 1993. From 1990 to 1993, he was with the Center for Electronic Packaging Research, Department of Electrical and Computer Engineering at the University of Arizona and worked as a graduate research associate. He is currently a senior engineer in the Advanced Packaging Development Department at LSI Logic Corporation, Fremont, Calif., where he has been invloved with the characterization and design of advanced IC packages for high performance ASIC applications since 1993. His research interests include signal integrity, electronic packaging and various VLSI CAD related topics. Dr. Hsu is a member of the IEEE.



Jerzy W. Rozenblit is an Associate Professor of Electrical and Computer Engineering at the University of Arizona, Tucson. He holds the Ph.D. and M.S. degrees in Computer Science. He specializes in knowledge-based system design, artificial intelligence, and simulation modeling. His principal research activities focus on the development of expert, computer-aided environments for design support. Dr. Rozenblit has been a US Army Summer Faculty Research Fellow. He has consulted for Martin Marietta, Semiconductor Research Corporation (SRC), and Siemens, and has been a reviewer for the National Science Foundation, Research Councils of Canada and Australia. He serves as an Associate Editor of ACM Transactions on Modeling and Computer Simulation. His research has been supported by NASA, Siemens, McDonnell Douglas, SRC, and the National Science Foundation. In 1993/1994 academic year he was a Fulbright Senior Scholar in Austria.