SRC TR 87-6

Computer Aided Printed Circuit Design

by

B.T. Sawyer and M.G. Pecht

70 ...

COMPUTER AIDED PRINTED CIRCUIT DESIGN

B.T. Sawyer and M.G. Pecht
Mechanical Engineering Department
University of Maryland
College Park, MD, 20742

ABSTRACT

This paper presents some of the engineering and programming developments which have resulted from the University of Maryland RAMCAD (Reliability and Maintainability using Computer Aided Design) effort for printed circuit boards (PCBs) and wiring assemblies (PWA)s. The paper overviews some of the problems associated with the PCB design process, examines the present status of computer aided PCB design, and then discusses the development of the University of Maryland RAMCAD system and the methods used to program for goal oriented design.

INTRODUCTION

The design of printed circuit boards (PCB)s, printed wiring boards (PWB)s and, more generally, printed wiring assemblies (PWA)s presents a large number of interesting and challenging problems [1-4]. In this section we present some of the general issues that guided our efforts in creating the University of Maryland (UMd) computer aided PCB system for reliable and maintainable design (RAMCAD).

PCB's are becoming more complex as technologies are developed which allow the construction of more densely populated boards with increasingly more sophisticated components. Methods developed to handle a few dozen designs a year are not adequate to handle a few hundred significantly more complex designs in the same time frame, if at all. The situation is analogous to the state of PCB manufacturing before the advent of wave soldering techniques. Hand soldering was adequate when the demand for PCB's was relatively low. Now it is impossible to fulfill batch demand without wave soldering. Yet, no corresponding "leap" has been made in the design process. It still relies on the equivalent of hand soldering techniques.

Designs are not only more complex but they must be more reliable. The demand for higher reliability of more complex designs places a double burden on design engineers. For example, even though a component reliability which may individually be considered acceptable, when a large number of such components make up a system, the total reliability can go way down. That is, the functionality of a system depends on the reliability of its subsystems. Failure of a subsystem may have caused inconvenience in the past. Now the same failure can cause catastrophic losses.

The design process can no longer be treated as a series of single variable problems based on meeting a required performance. Design engineers must include reliability, maintainability and supportability up-front in the design process. In addition, functionality, operating conditions and manufacturing procedures must be taken into account. This type of unified life cycle engineering approach to design, places additional demands on an already small number of expert designers who have no comprehensive design theory to support them [2].

COMPUTER AIDED PCB DESIGN

The basic difficulty with developing a PCB CAD system is that there is not a well defined theory to describe the process. Computer solutions to problems require a level of specificity that design theory does not provide. This helps to explain why there are many CAD tools which perform specific design functions and analyses but few if any which integrate these tools into a single system. For example, there are theories which describe the processes of determining thermal reliability, mechanical stress, component placement, routing, etc., but none which describe the interactions between these often competing design procedures.

The use of a single acronym (CAD) for programs that are used in design work can lead us to overlook the fact that these programs typically use a variety of distinct programming methodologies. Even though they bear the same label, all CAD programs are not alike. The method used in a given program depends on the nature of the process being modeled, the background of the programmers writing the code, and the institutional structure in which the programs are developed.

The computer aided design of a PCB involves a number of different capabilities. Some of the most important of these are listed below.

Computation: This is perhaps the most developed aspect of CAD. Well defined models for calculating temperatures, stresses, reliability, etc. are available and are not difficult to translate into computer programs.

Graphics: Graphic displays present a complex design in a format that is easy to grasp and manipulate. Programming for graphics is distinct from computational methods even though computations are important for graphic displays. The utility of this object oriented methodology has spurred dramatic developments in hardware, software, and graphics theory. In fact, CAD and graphics are generally treated as synonymous.

Expert Systems: The demand for integration of competing design goals has led to the application of artificial intelligence methods to CAD programs. Further, the lack of expert design engineers necessitates the development of systems which make expert techniques available to designers who have not had the time and experience to become true experts. The programming methodology for expert systems is not as well developed as computation or graphics. Reasons for this will be discussed later.

Database management: The variety of data types involved in a PCB design system requires good data managment methods. This area of computer science is well developed and provides a number of methodologies from which to choose. There is, however, a weakness in some database management techniques when they are combined with systems which require artificial intelligence capabilities. In particular, hierarchically defined databases often lack the flexibility that AI methods require.

The programming model for a design system must unite computational, graphic, expert system, and database methods into a coherent whole. This is not a unique discovery, but it leaves open the question of how this integration can be achieved.

A DATA CENTERED MODEL

We were faced with the task of developing expert control and decision support for a system whose nature is not clearly understood. In the face of this challenge we studied the PCB design process and noted that there are a finite number of parameters which can be configured to reflect the state of the design. Furthermore, it appears that present and predicted future requirements for a design can be expressed in terms of those parameters. As a result, we decided that an expert system could be used to monitor, control and aid in decision support of the design process by comparing specified goal states to the state of the data.

The "data centered" model provides a foundation on which we base our research activity. The programmers focus on data types and structures and clearly define the interaction of the data. The engineers examine the data for correctness and define the nature of the processing activities. As the system develops, guidelines for optimization and trade-off studies are incorporated into the expert system and design data states are compared with the goal states.

The method we chose to develop an expert system for design requires close cooperation between programmers and engineers. Without this interdisciplinary effort we doubt that a workable model can be developed. Futhermore, we found it essential to include actual analysis modules as part of the development of the expert system controller. It was not until the analytical data was generated that we could judge the effectiveness of the controller and could add "intelligent" guidance procedures to the system. In fact after a number of analyses modules were complete we saw that the controller and the data had to be restructured to accomodate more than trivial trade-off studies.

The data centered system permits modular development of system functions. The modular approach is integral to our model building. It allows a number of persons to work independently on analysis and control modules. More importantly, since all the requirements of a PCB design system are not known, we require the capability to add and modify system functions without affecting those that are operating adequately.

Modular development can also be applied to the system controller. Many expert controller models treat the system as one composed of many independent processes. At the same time they consider the controller to be monolithic even if it is composed of individual program modules. The data centered model does not require an omniscient controller. On the contrary, since we are working with a process whose attributes are not all known, it is necessary to modularize the control processes. We thus created modules to monitor the attributes we have identified and add others as new discoveries are made.

In our system we have separated the functions of process control (scheduling) from design control (goals). We have further refined the goal oriented functions into distinct modules which may act as independent units. Actually, the goal oriented modules work in pairs. One module is used to establish the design goal in a specific area and another module checks the data relative to that goal. Once the characteristics of a particular trade-off study are defined we can introduce a module to mediate among competing design goals.

DATA HANDLING

Before writing any code for our system we examined the design functions that were to be incorporated and then carefully defined data structures which contained the necessary parameters. It is important to distinguish between structures that are well defined and those that are rigid. Although our structures have a clear definition they have flexibility built into them. Initially we limited the number of analysis functions and their associated parameters in order to study the results produced by this prototype before defining a more general system.

In the Maryland RAMCAD system we adopted a functional approach. That is, the system is viewed as a series of integrated functions rather than a group of program modules. As the system was implemented we created the program modules we felt necessary to perform a given function. In some cases, one module performs several functions. In others, two or more programs perform a single function.

Along with the definition of data structures we had to decide on a method of control and communication among independent modules. Since we were using a data centered model we chose the local database as the control and communication medium. The local data base consists of all the files which contain design data, system control data, and data provided by analytical modules.

An "active" file consists of a record which contains the PCB name and a number of boolean fields, one for each module. When a module completes its operation on the data, its boolean field is set to "true" so that the system controller can determine what has been done to the design. The "active" file's record is extracted from a file of all the designs in the system. This file is updated after each function is completed so that the present state of any number of designs is maintained by the system.

A component file consists of records for each of the board's components. Each record contains all of the characteristic constants for that component as well all values which are calculated by the various modules. Constant fields include such parameters as rated power, number of pins, resistances, Pi factors, and weights. Calculated fields include case and junction temperatures, failure rates, and temperature dependent Pi factors such as environment and voltage stress.

The decision to use such a component record has advantages and disadvantages. The major disadvantage is the difficulty of using a single record structure for the various types of electronic components. A large (250 component) board requires about 60K of disk storage and 10K for the file of sorted keys.

The advantage of keeping all necessary working data in a component record is that each program module gets all the data it needs to perform its task, without having to rely on external "look up" files. A limited number of modules aoperate directly on the local data base. Most modules have independent data structures tailored to their needs. A programmer is thus free to manipulate the data in whatever way is necessary to perform a given task. A test of a program's correctness is made by examining the data it produces. More importantly, the component file becomes the means of communicating data from one module to the next. For example, one module calculates junction temperatures which are transferred to the local data base. Another module uses these temperatures to calculate failure rate and so on. Just as the active file contains the state of the design, the local data base component file contains the state of each component in that design.

A board file is used to store data which is characteristic of the PCB as a whole. This includes environmental temperature, allocated mean time between failures, natural frequency, failure rate, and total power dissapation. The board file, like the component file, stores data generated by a module.

In addition to the active, component and board files there are files which contain graphics and placement data, program constants, menus, labels, and MIL-Standard specifications. These files are used by particular modules for their specific needs.

Besides communicating the data and the design state, the local database is used as a source of control information. A program module we call "Mother" performs the functions of user interface and scheduler of system events. Another module, the program executive, executes the modules that Mother schedules. These two modules communicate by means of a file which contains the names of the modules to be executed. Mother creates this batch file and the executive executes the modules whose names are in it.

The batch file method for process control gives RAMCAD its versatility. Program modules are created independently and do not have any system control functions. They do a specific task, leave the results in the active, component and board files, and then exit. All process control resides in one module (Mother). This is in contrast to the CHAIN command which "locks in" the order of module execution. Another important result of the batch file approach is that modules may be written in any compiled language. The CHAIN command is normally limited to executing files of the same language.

GOAL ORIENTED DESIGN

The data centered model uses the local or working database as the source of all monitor, control and communication information. One file holds the history of the design and another the results of the latest analyses. A third file is to be the source of scheduling data for consecutive processes. Other files are designated to maintain the history of all system designs, supply program constants, provide graphics data, etc.

A module called "Mother" was created to oversee the system. This module is the user interface which provides system level menus and the controller which schedules all other modules. A separate program was written to execute the sequence that Mother writes to the schedule file. This module had no control function of its own.

The need to have a system that accepts design goals and checks the progress of the design against these goals led us to develop modules which provide goal orientated control. Selection of a performance environment is key to reliability prediction. Then based on user allocated reliability measures, feedback is given on the extent to which the design meets the allocated value. Modifications of that value can then be made.

The goal oriented monitoring technique can be used for any number of desired outcomes. It is ideal for a system which must guide the designer when multiple goals must be met. As we become aware of other suitable goals and can specify them relative to our database, the appropriate set-up and monitor modules will be added to the system.

The system controller does not have to be reprogrammed for each new goal. All it needs is the order of execution for modules (Figure 1). For example, the sequence for determining and monitoring the MTBF is: Parts Stress is executed before any thermal analysis and Reliable follows every thermal module. The details of the goal are not part of the system controller. They are taken care of by the set-up and monitor modules. This method of goal oriented control allows us to add any type of goal simply by specifying the order of execution of modules. The details are taken care of by the modules. Design goals can be automated by changing the set-up module to access default values from the database.

In all instances of monitoring goal specifications, the modules are executed in a background mode. They become visible only when necessary and do not interfere with the system sequencing. That is, a module informs the designer that a goal is not being reached and offers suggestions for design improvements, but does not prevent continuation of the design process. Our monitor functions are written to inform and suggest corrective steps. They do not assume control or force corrective action.

The monitor function is based on an expert system model. The state of the data at any time may be examined and compared to a goal state using its function. The advantage of the monitor function in a design system is that it frees the designer from the task of keeping track of a large number of design parameters and an even greater amount of data. It provides the assurance that design goals are being monitored and promotes creativity by removing some of the tedium of the design process. The monitor uses expert system methods not to solve the problem of creating a successful design but rather to act as an advisor to the person who is ultimately responsible for the final product.

PROGRAM INTEGRATION AND EXPANDABILITY

It is possible to incorporate within a single program structure, every aspect of the design process, from parts selection to the manufacture of the final product. However, because the rapidity of change in the electronics field exceeds the ability of programmers to write and verify the code to produce a reliable product, and because there already exist excellent programs for routing, placement, schematic capture, thermal analysis, reliability analysis, and manufacturing control, a modular program structure with means of communicating with other systems appears most fruitful.

There are barriers to this approach of sharing data. Most notable is the reluctance of organizations to "give away" the results of many thouands of hours of work. Even if contractual arrangements are made to share data, there is still the problem of incompatible data types, interfaces, and programming conventions. A government approved data transfer standard would help overcome this obstacle, much the same as the definition of language conventions helped to make software transportable among different machines and operating systems.

The Maryland RAMCAD system uses the data centered approach with specialty translators to access programs written in various languages and data formats, Even within the system, the translation paridigm is used for most modules. This simple method allows us to incorporate otherwise incompatable modules into the system to change rules to add parameters without affecting the operation of the system. This principle can be used to incorporate programs that perform functions that are superior to what we could develop in any reasonable time frame.

CONCLUSIONS

The science of second level electronic packaging design is an area which has not as yet formulated theories to guide those in charge of producing the quality and quantity of PCB's demanded by our society. Expert designers have always been scarce. Now the sheer volume of work exceeds the ability of the educational and industrial communities to produce the number of experts required to fill the demand.

A system used to increase productivity must be relatively simple to use and must not require radical changes in the organizational structure of the company it is meant to help. A design system must be adaptable to the organizational structure in which it is used. An organization with a large investment in hardware and software has good reasons for continuing to use what they have rather than switch to a different method of solving their design problems. It is likely that if a system is used for any length of time it colors the organizational structure to such an extent that changing the system causes dislocations in the organization itself. Adjustments are always required when a new or different system is put in place but radical adjustments are seldom acceptable. Even a superior system will fail if the people using it are alienated by it.

A completely automated PCB design system is not considered practical at this time. This is due to the fact that the overall design process is not well understood and because criteria for a successful design differ depending on the experience of designers and organizations involved in producing PCBs. It is more realistic to allow a particular designer or organization to establish the design goals and then to make decisions when presented with data on competing goals. The data centered model gives us the ability to monitor and handle design rules and guidelines, and can suggest to the designer ways to interactively improve the PCB design.

ACKNOWLEDGEMENTS

This research was supported in part by the Institute For Defense Analyses and by the Systems Research Center under grant

No. CDR8500108 given by the National Science Foundation.

REFERENCES

1. J. Horan, M. Palmer, M. Pecht, and Y. Wong, "Intelligent design of printed wiring boards", Proceedings : Association for Computing Machinery, Vol. 1, pp. 123, (1985).

2. J. Naft and M. Pecht, "A RAMCAD/ULCE workstation structure", Proceedings : 1987 Annual Reliability and Maintainability

3. M. Pecht, J. Naft and M. Palmer, "Workstation requirements for printed wiring board design, Proceedings : IEEE Workstation Technology and Systems, Vol. 1, pp.63, March (1986). Symposium, Jan (1987).

4. M. Pecht, M. Palmer, W. Schenke, and R. Porter, "An investigation into PCB component placement tradeoffs", submitted to IEEE Trans Reliability, Aug (1986).