Lehigh University
Lehigh Preserve

Theses and Dissertations

1989

Automation of IC package design using PEX /

Ravi K ulasekaran
Lehigh University

Follow this and additional works at: https://preserve.lehigh.edu/etd

Part of the Manufacturing Commons

Recommended Citation
K ulasekaran, Ravi, "Automation of IC package design using PEX /" (1989). *Theses and Dissertations*. 4966.
https://preserve.lehigh.edu/etd/4966

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.

AUTOMATION OF IC PACKAGE DESIGN USING PEX

by

Ravi Kulasekaran

A Thesis
Presented to the Graduate Committee
of Lehigh University
in candidacy for the Degree of
Master of Science
in
Manufacturing Systems Engineering

Lehigh University

1988

This is to certify that this thesis has been completed as a requirement of the Master of Science Degree in Manufacturing Systems Engineering.

October 5, 1988

Date

Thesis Advisor

Program Director

Department Chairman

// ACKNOWLEDGEMENT

I would like to thank the help and cooperation of my colleagues in the SRC project. I would like to thank Dr. Donald Hillman, my advisor, for his support and encouragement throughout and after the project. I would also like to thank Dr. Richard Decker for steering the project and providing useful advice. My special thanks to Bob Voros.

TABLE OF CONTENTS

1. Abstract 1
2. Semiconductor Research Corporation (SRC) Project 2
3. Packaging Trends 3
4. Packaging Requirements 4
5. General Design Guidelines 7
6. Expert Systems 10
7. Knowledge Acquistion 12
8. Knowledge Representation 13
9. Knowledge Application 18
10. Generating Explanations 18
11. Logic Structure 19
12. Rules of Thumb 20
13. Financial Implications 21
14. Limitations of Expert Systems 22
15. Prolog 23
16. Structure of Turbo Prolog 25
17. PEX (Packaging Expert) 29
18. PEX Structure 30
19. Bibliography 32
20. Appendix A 34
21. Appendix B 43
22. Vita 52

LIST OF FIGURES

Fig 1. Interconnections tolerance vs cost 6

ABSTRACT

This paper describes an expert system PEX developed for the Semiconductor Research Corporation (SRC) for designing packages for integrated circuit chips. It was developed using Turbo Prolog and the knowledge supplied by members of the SRC. The paper discusses the requirements and the role of an expert system for packaging in the future. The paper also takes a look into the development and requirements of an expert system. The expert system was part of a larger project for the SRC.

The development of this expert system with respect to the systems point of view is looked into. The concepts in modern day manufacturing strategies are tied with the use of efficient tools like the expert system. The paper puts into perspective the need for systems thinking in a multi-disciplinary approach such as IC chip packaging.

The knowledge representation schemes used for development of expert systems are discussed. The merits and demerits of each scheme are analyzed. Finally, there is an overview of Turbo Prolog.

SEIMCONDUCTOR RESEARCH CORPORATION (SRC) PROJECT

In the past decade there has been tremendous advances in the field of semiconductor packages. The emphasis has been on putting more number of circuits and reducing the size of the silicon die. The industry has moved from large-scale integration (LSI) to very large-scale integration (VLSI) of IC chip design. The increased performance of the silicon die alone is not sufficient. The overall performance of the die along with its package in the environment is a crucial factor. Thus, the design of packages has to keep pace with the rapidly changing technology of the silicon die. There are serious limitations when it comes to the packaging of the silicon die.

In order to be competitive most companies have to design and manufacture their chips with a low turn around time. The pressures on the packaging engineer could lead to a proliferation of package designs and is time consuming. In order to optimize the work of a packaging engineer and let him/her be involved with other intricate packages, the idea of developing an expert system that could handle the lower end of the package spectrum was envisaged.

The multi-disciplinary approach of package design makes such a system very useful. Equiping the packaging engineer with other associated tools like CAD and finite element analysis packages will help in reducing the time taken for design.

PACKAGING TRENDS

In the past decade, the semiconductor industry has focused in increasing the performance and density of the semiconductor components. One tends to forget that the overall performance of the chip does not solely depend on the silicon die alone. The silicon die needs outside connections to be able to function. It needs a mechanical structure to mount it to protect it from hostile environments. Its package must be able to cool it as it dissipates heat to perform its function. Further, the package must be easy to attach to other system-level interconnection units. Thus far the design of the silicon die and its package had been independent pursuits, leading to a proliferation of chips of various dimensions.

On the one side, the silicon manufacturer must package the die in order to protect it, test it, and deliver it [Packaging Trends]. The silicon manufacturer is interested in the lowest-cost (to him) die package. Its use in a "system" environment is not under his control and not a major concern. To a degree, the silicon manufacturer is responsible to provide a means for cooling the die. On the other side, the systems manufacturer controls the mounting and removal environment, the power distribution, the cooling and the space available. The non-vertically integrated silicon manufacturer's technology base does not usually include all the available possibilities. The result is a compromise which is the reverse of the ideal situation. There have been many instances where the use of a slightly more expensive component package will more than pay for itself by costs saved elsewhere.

The above mentioned reasons indicate that the situation is ideal for "systems thinking and integration". The design of the silicon die and the package should go hand in hand and reduce the proliferation of many standards and products. Systems manufacturers should study the whole environment from die to system packaging and join with the semiconductor manufacturer to choose component packages that minimize the total packaging cost. The time-honored practise of using the least expensive package as chosen by the silicon die manufacturer must go. In particular, "system cooling" techniques and "system-level interconnections" and "system temperature environment" must all be considered in package designs.

PACKAGING REQUIREMENTS

The package for a VLSI die connects the die bonding pads, which are located on approximately 0.008" centers, to an intermediate dimensional set on 0.020" to 0.050" centers on the lead frame internal to the component package and finally steps the dimensions up to 0.100" center pins which match PCB dimensions.

The attaching of the die to some type of mechanical substrate to hold it stable to allow lead attachment and to allow heat conduction away from the die and finally to permit the sealing of the die from hostile environment is needed. The die sizes typically range from 0.050 mils to 0.300 mils square with lead bond pads on 0.008" to 0.010" centers. The potential footprint of the die package can be one-half to one order of magnitude larger than the silicon die.

The ideal interface between the system and the VLSI package would provide the following features:

- Easy attachment
- Easy removal
  - no component damage
  - no system damage
- Small footprint area
- Minimum interconnection interference
- Adequate cooling
- Easy cut and jumper design changes
- Low capital investment for attachment/removal tools

Other desired features would be low-cost, zero-insertion force, reliable, low-profile socket.

Silicon component manufacturers are not generally concerned with system level interconnection costs. They tend to minimize the cost of the component package irrespective of its system implications. PCBs are still the dominant means of achieving system level interconnections. As fig 1 shows the cost of PCB. A very high volume, two layer, low tolerance PCB costs

about $0.20 a square inch. A reasonable tolerance, high volume, 10 layer board costs $3.25 a square inch. A 10 layer board with special close tolerance costs over $10.00 a square inch. Compounded with the cost is the issue of availablity. As the number of layers increase and tolerances get tighter, the number of available vendors decreases very rapidly. Most of the PCB manufacturers find it more profitable to produce high volume, relaxed tolerance, two to four layer boards than high tolerance, low volume , complex layered boards. The lead times for the later type of boards is greater than that for the former. The PCB system cost should be a consideration while selecting a custom silicon chip supplier and his proposed package for the chips.

In order to reduce the interconnection complexity at the PCB level, a top-down system design which is interconnect dominated can be used to specify where each component lead will be located on the component package. Also, the package mounting method to the PCB should be carefully chosen.

The process of determining whether any available package is acceptable is similar to that of specifying a new design. With higher lead count chips, there are problems associated with the footprints. As a result, any package chosen will be very application sensitive and must be designed from the die out to provide the best total solution for system requirements.

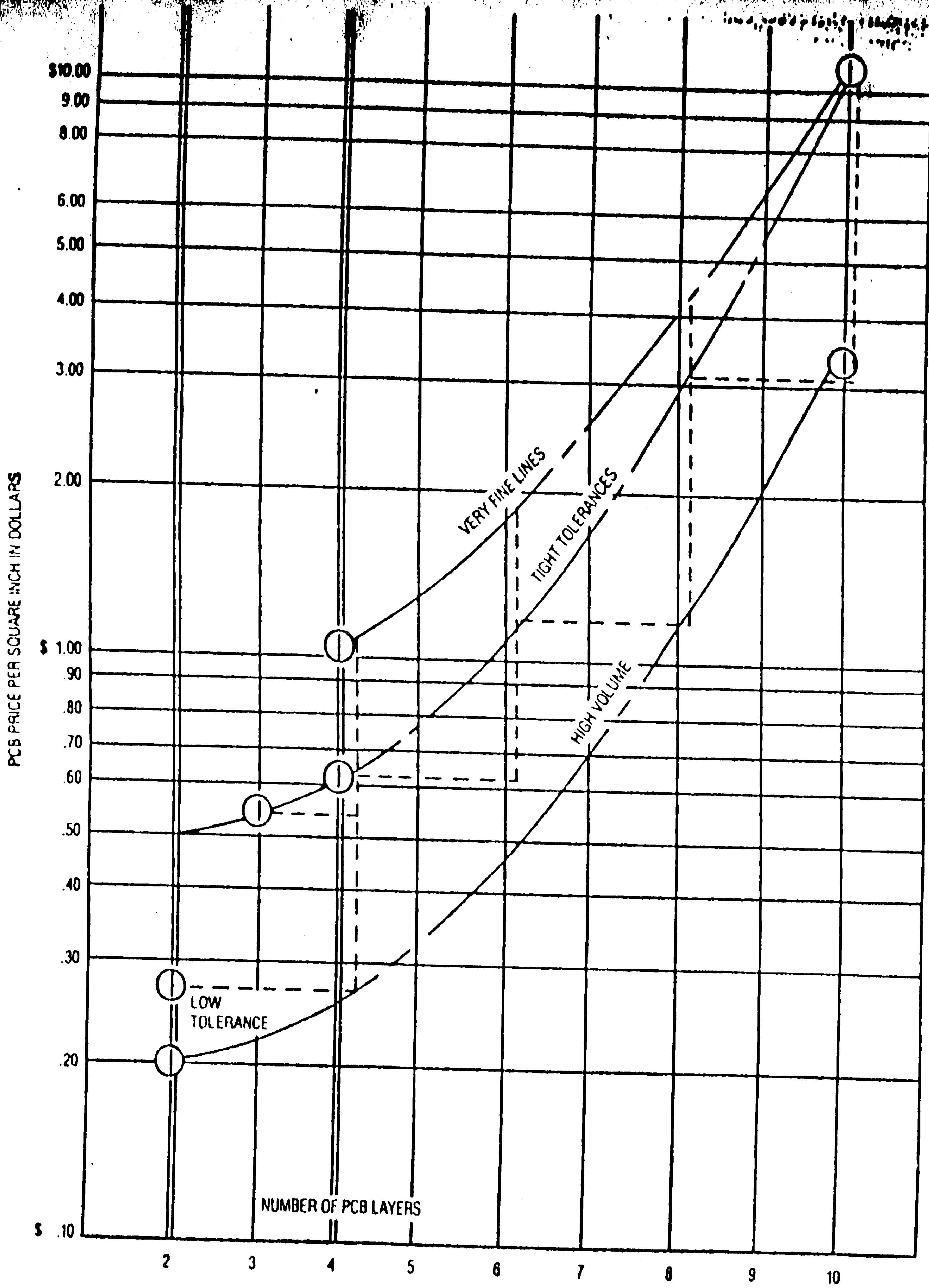

fig 1

complexity vs. cost

GENERAL DESIGN GUIDELINES

The Physical Housing

Cavity Size

VLSI die design may range from 0.100" to over 0.300" on a side [Nelson]. The cavity selected must be large enough to fit the largest die in a given die family. However, if a wide range in die sizes is to be accommodated, there is a potential problem of excessive wire bond lengths for the smaller dice in the family. Each company has developed their own rules for cavity size depending on the established die attach process.

Cavity Flatness

The larger die in the VLSI families require cavities approaching 0.5" square. VLSI die attach yields and thermal considerations require that cavity camber or flatness be held to 0.003" maximum with 0.002" preferred for eutectic attach.

Cavity Depth

With the larger wafers, dice are getting thicker. Some wire bonding processes require that the wire bond ledge be level with or higher than the the die bonding pads.

Outer Wire Bonding Ledge

VLSI die may require a large number of bonding pads in a relatively small area. As a result, the die considerations alone may not require a cavity large enough to provide space for the outer lead bond pads.

Cavity Sealing

Most VLSI packages provide ample space for the sealing area because of the real estate required for the I/O pads.

Package Electrical Environment

Resistance

Resistance is not to difficult to handle because the conductor properties of packages are well know and thus permit one to make reasonably accurate calculations.

Capacitance

The geometries of the conductors after lamination are not precise and the dielectric properties of multilayer constructions are difficult to characterize. By controlling conductor length, ceramic layer thickness and proximity to planes or other circuits there is limited flexibility in controlling capacitance.

Inductance

Inductance specifications require minimizing of bonded wire lengths as well as general package conductor lengths and widths. Parallel paths to minimize ground inductance can be designed with advanced planning.

Package Thermal Environment

The package should provide a cavity orientation that points the die backside towards the air flow.

The package construction and die attach should minimize the junction to case thermal

impedance.

The type of package lid selected is also dependent on the heatsinking method.

Wire Bond Ledge Length

The bond pad must be long enough for tool clearance. A 30 degree wire bond wedge is the most critical. To accomodate one the following approximation can be used:

L = 2H + 0.010". Where H is the thickness of the top layer of ceramic that must be cleared by the tool.

Package Interface Design

The following points must be kept in mind:

List corporate preference for printed circuit board pad width, hole size, line width and spacing.
Is impedance control required? What value?
Are all package terminals required to be accessible for probing?
Are cut jumper wire design changes to be accomodated?
Is the connector to be used? What are its requirements for spacing, hole size, etc.?
If a connector is not used, the package to board attachment pattern and "fanned out" vias establish the footprint.
Check the PC board power density versus the system cooling capability.
Establish cost and performance models comparing component mounting densities with PC board layer requirements.
Share the tolerances so each packaging element is producible and available at an equitable cost.

EXPERT SYSTEMS

An expert system is a computing system capable of representing and reasoning about some knowledge-rich domain with a view to solving problems and giving advice [Jackson]. Most commonly, they are based on "if/then" rule, which in conjunction with facts about a particular subject can logically progress through a given problem to arrive at an appropriate solution [Davis]. One of the their main attractions is that they often permit the actual users to have a say in the way the systems operate. Such input is usually necessary because expert systems are meant to mimic the knowledge and procedures of real people. It is a computer system designed to simulate the problem-solving behavior of a human who is expert in a narrow domain [Denning]. As a result, applications programs tend to be much closer to users' actual needs than traditional software programs written in isolation by data processing professionals.

Expert systems can be accessed by many users at a time -- a practice not feasible with human experts [Davis]. The expert system stays with the organization that owns the system, even if the human expert leaves, and thus becomes a permanent asset. An expert system is a restricted form of a system of logic. It attempts to compute a sequence of strings representing the steps in the solution of a problem. The sequence serves as a proof of the solution.

It can be distinguished from other kinds of artificial intelligence programs in that [Jackson]:

- It deals with subject matter of realistic complexity that normally requires a considerable amount of human expertise
- It must exhibit high performance in terms of speed and reliability in order to be a useful tool
- It must be capable of explaining and justifying solutions and recommendations in order to convince the user that its reasoning is, in fact, correct

Expert systems rely on the links between :

- Knowledge acquisition
- Knowledge representation
- Knowledge application
- Generation of explanations

KNOWLEDGE ACQUISITION

Knowledge acquisition is the transfer and transformation of potential problem-solving expertise from some knowledge source to a program [Jackson]. This is the most important step in the development of an expert system. It is also the most time consuming process. The knowledge engineer should be in a position to ask pertinent questions and gather information that will be useful in encoutering situations that are atypical. The knowledge engineer should be able to pick the brain of the expert and collate the information into a logical pattern for the development of the expert system. The success of the expert system will be determined by the knowledge acquistion process. The knowledge engineer should be able to perform the task of the expert in solving a problem by going through the knowledge gained from the expert.

KNOWLEDGE REPRESENTATION

Knowledge representation is concerned with the way in which information might be stored in the human brain, and the ways in which large bodies of knowledge can conveniently be stored in data structures for the purposes of symbolic computation. Knowledge is the symbolic representation of aspects of some identified domain.

There are primarily three ways of representing knowledge [Denning].

- logic programming system
- rule-based system
- frame-based system

Logic Programming System : It embodies explicitly the model of proof in mathematical logic. Rules are represented in the form of

q :- p1,....,pn

interpreted as : If predicate p1,....,pn are all true, then the predicate q follows. The principle of transitivity allows us to replace a pair of rules,

Q :- p1,...., pn
s :- Q, r1,......., rm

with the new rule

s :- p1,......., pn, r1,...., rm

This pattern of reduction is the basis of resolution, for finding the ultimate consequences of given conditions. It is not necessary that the Q on the left of the first rule be the same as the Q on the right of the second rule; it is only necessary that they can be made the same by substituting for some of the variables.

Rule-Based System : Its database is structured as a hierarchy of rule sets. Each rule set comprises a series of condition-consequence rules as defined earlier. Two strategies are common for evaluating the consequences of the initial conditions given by the user. Under forward chaining, evaluation proceeds by tracing rules from left side to right, until the ultimate consequences of the given conditions have been found. Under backward chaining, evaluation proceeds by tracing rules from right side to left, until the ultimate antecedents of a hypothesis about the given conditions have been found; hypotheses that conflict with the given conditions are discarded along the way. The method to be adopted depends on the problem. If much data is available, it is often faster to chain backwards from a few likely hypotheses. If there are many possible hypotheses, it is usually better to chain forward from the limited initial data.

Frame-Based System : A frame-based system is centered on a hierarchy of descriptions of objects referred to in the rules. The description of a class of objects or an individual object is called a frame. The relation "is an instance of" organizes the hierarchy of frames. Rules are used to describe functional relationships among frames. Any property of an object that can be derived by locating it in the hierarchy of frames need not be explicitly represented in the rule sets in exchange for complexity of the catalogs of frames and of the inference algorithm.

The different knowledge representation formalisms are:

- **Natural Language**
- **File/ Record structures**
- **Formal Logic**
- **Production rules**
- **Slot and filler formalisms**

Natural Language

This is the most powerful of all knowledge representation schemes. It needs reference to the context to resolve ambiguity. Syntax and semantics are not fully understood. The sentence structure is non uniform. These factors make it difficult to use natural language for systematic knowledge representation.

File/ Record Structures

Records

A record is a collection of one or more named fields each of which contains the symbolic representation of some aspect of a domain. A record represents a single entity in some domain. It is efficient in knowledge processing since it is uniform. However, the fixed format, difficulty in modification of records and limited applications of the record structure are the disadvantages.

Conventional Files

The structure is a contiguous collection of records which are all of the same type and are related in some way. Conventional file systems are best used to represent one-many relations rather than many-many relations.

Records linked by Pointers

The structure is in the form of a linked list. The pointer indicates the begining of the next record. These pointers take up space. Records are usually physically distributed, with the result that several types of processing tasks like computing averages are inefficient. Moving a record from one physical address to another requires updating the pointer fields.

Relational Approach to Knowledge Representation

The knowledge is represented in the form of flat files. The advantages of this form of representation is is its well-developed mathematical theory of relations to handle operations and the ease of expanding the knowledge base.

Production Systems

A production system consists of a set of rules, database management system and a rule interpreter. The rule is in the form of an "if-then-else" statement. The database management system manipulates a body of relevant factual data. The rule interpreter identifies applicable rules and determines their order of application. The advantage of a declarative approach over a procedural approach is the ease of addition and modification of knowledge.

Slot and Filler Systems

Production systems are not very good in representing the structure of a knowledge domain. They do not offer much help in organizing all the different aspects of a given entity into a coherent framework. The slot and filler approach overcomes this problem. Different types of slot and filler representations are frames, semantic nets, scripts, conceptual dependencies. In frames, all assertions about a particular entity are held together. Frames can then be linked into frame structures.

KNOWLEDGE APPLICATION

This relates to the issues of planning and control in the field of problem solving. Expert systems design involves paying close attention to the details of how knowledge is accessed and applied during the search for a solution. Knowing what one knows, and knowing when and how to use it, seems to be an important part of expertise; this is usually termed 'meta-knowldege'.

GENERATING EXPLANATIONS

The whole issue of how to help a user understand the structure and function of some complex piece of software relates to the comparatively new field of human/computer interaction, which is emerging from an intersection of artificial intelligence, engineering, psychology and ergonomics.

LOGIC STRUCTURE

One of the benefits of an expert system is its inherent logical flow [Teschler]. Expert programs are organized into rules and facts with a structure that is easy for non-programmers to grasp. This is important when working with individuals who are unfamiliar with programming. People serving as sources of expert information can often look at the program code and understand the intent of the rules and facts, simply because the information is expressed in ways that resemble English sentence structure. This enables the human expert to examine program code and see at a glance whether it expresses the right rule of fact.

The logic structure of expert programs is also such that programmers can come up to speed quickly when joining a project. This is especially good in large software efforts where personnel may turn over several times. Similarly expert programs are often easier to maintain and change than software written in languages such as Fortran or Basic.

In all, many problems tackled with expert systems could be handled with conventional programming techniques as well. But the branching and decision trees needed in ordinary languages are often more difficult to implement.

Typical uses for expert systems to date include situations where the expert program substitues for service manuals or instructions. In applications currently under development, such software serves as a smart business form by guiding users through instructions. Programs check entered answers for reasonableness and consistency and verify entered data to make sure they do not contradict other informtion entered elsewhere on the form. One expert program implemented by IBM's San Jose plant diagnoses 98% of the faults in disk drives. The software replaces about 30 lb. of manuals that guide technicians through the diagnostic process.

The term expert program is actually a misnomer. Most expert systems fielded today provide advice that is not really at an expert level. At best, such programs give guidance that allows rank amateurs to be mediocre performers. Many expert programs fuction simply as training and productivity aids. Thus, a rule of thumb is that expert programs make sense where less than optimum performance is acceptable.

RULES OF THUMB

Some of the rules of thumb in the development of an expert system are:

- Narrow uses are best:- Programs that solve well focused problems are the easiest to generate. Diagnostic programs are relatively easy to produce if the end result is the recommendation or selection of a single best alternative. If diagnosis results in a list of several possible candidates that must be evaluated, the program may need complicated rule-searching techniques.
- Check interfaces:- Expert systems are increasingly used directly with spreadsheet and database files.
- Get a feel for size:- Try to sketch the logical relationships of the rules and facts in the application. This will give an indication of program size while revealing redundant rules and missing information.
- Beware of jerks:- Expert system developers must work with a human expert to obtain the knowledge which the program is based. If the human expert and the programmers do not communicate effectively with one another, the resulting program may be disappointing. Interpersonal skills are very important for the participants in the development of the expert system.

Expert systems are merely sophisticated pattern matching programs. To a computer, the rules and facts in an expert system program are meaningless patterns that can be matched to trigger specific outputs. Present day expert programs are brittle because they cannot tell when they are operating outside their small domain of expertise, or when they are given irrelevant information that will lead to a ridiculous conclusion.

FINANCIAL IMPLICATIONS

Most application programs sold are highly specialized [Freedman]. At the moment, several factors restrain growth in the applications market. Programs are expensive because of high development costs. A company usually finds it has to spend additional money on consultants and in-house programming efforts to install and fine-tune the program. In addition, expert systems typically run on symbolic-processing computers, which generally cost $40,000 to $100,000, for a system that supports only a few users, and they cannot easily be tied into conventional software and databases. According to Neal Goldsmith of The Gartner Group (Stamford, CT) there is yet no expert system that can simply be bought, plugged in, and used to automatically access the corporate database and then run.

LIMITATIONS OF EXPERT SYSTEMS

The most important limitation of an expert system is its reliablity [Denning]. This is because of the limitation of the information put into the database. They can tackle only those situations which are implicitly in the databases. The trial and error process of knowledge engineering and testing can produce inconsistent and incomplete databases leading to gaps in knowledge when unexpected situations are encountered. Construction of large databases must be modular to facilitate maintenance and modification. The problems with testing of expert systems is difficult. The correct behavior of the system in a particular situation could be subjective. Expert systems can be used as intelligent assistants with the supervision of humans. The best these systems can do is imitate human experts.

One key reason for the subpar performance of programs fielded thus far is that most solve problems by selecting solutions from a predefined list of possible candidates [Teschler]. Problems that these programs solve are formulated as diagnosis. In general the diagnostic approach can be appropriate when the problem at hand involves a finite number of strategies or solutions, each of which can be prescribed based on the existing conditions. Expert programs can give good advice if all possible solutions have been identified and entered in the program.

An expert program cannot suggest a solution that it does not know about. If a more creative plan, diagnosis, or answer can solve the problem better, the program will miss it. This is a significant limitation of the expert system technique that potential users and developers should understand. Another area where simple expert approaches are likely to suffer is in handling unanticipated problems. Most such programs employ shallow knowledge, predicting out-comes based only on the experience of the human expert who provided knowldege for the program. If the program runs into an unforeseen situation, it has little information to guide it to a solution.

Early successes with expert systems have lead people to have unjustified expectations of the technology. Without a good understanding of expert system limitations, it is hard to know where the expert system fails.

PROLOG

Prolog is a fifth generation language [Turbo Prolog]. It is declarative by nature. Given the rules and facts, Prolog uses deductive reasoning to solve the problem. This is contrasted by procedural languages that need explicit step-by-step procedures to solve problems. The user supplies the description of the problem (the GOAL) and the ground rules for solving it. The description has three components:

1. Names and structures of objects involved in the program
2. Names of relations which are known to exist between these objects
3. Facts and rules involving these relations

Examples:

Facts: Material is plastic

Rules: Package is non-hermetic if material is plastic

Deductions:Given the facts:

john likes mary
tom likes sam

and the rule

jean likes X if tom likes X

Prolog deduces

jean likes sam

The execution of prolog programs is automatically controlled. Prolog tries to find all values

that satisfy the goal. Prolog uses backtracking, i.e. once a solution has been found, assumptions are re-evaluated to see if new variable values will provide new solutions.

STRUCTURE OF TURBO PROLOG

The structure of a Turbo Prolog program is as follows:

```
domains /* declares types of objects */

        person, activity = symbol

predicates /* delcare type of relations */

        likes ( person, activity )

clauses /* describe facts and rules */

        likes ( ellen, tennis ).
        likes ( john, football ).
        likes ( tom, baseball ).
        likes ( eric, swimming ).
        likes ( mark, tennis ).
        likes ( bill, X ) if likes ( tom, X ).

goal:- likes ( bill, baseball )

reply: True
```

Variables

```
likes ( bill, X ) if likes ( tom, X ).
```

All variables start with upper case letter. It doesn't matter if other letters are of different case.

Anonymous Variables

car(_, _, Age, _, Cost)

will match

car(**chrysler, 8000,** 3, red, 12000)

car(**ford, 9000,** 4, blue, 24000)

where the

```
domains
    brand, color = symbol
    age, price = integer
    mileage = real

predicates
    car( brand, mileage, age, color, price )
```

Standard Domain Types

char	eg. 'a'
integer	-32768 to 32767
real	42, -86.72
string	"sam's book"
symbol	1. sequence of characters e.g. joan
	2. character sequence in double quotes e.g. "Joan"
file	begin with lower case e.g. files = destination

Lists and Recursion

Lists are fundamental data structures.

e.g. [tom, sam, jane, sally].

The head of the list is the first element in the list.

e.g. head = tom

The tail of the list is the rest of the list.

e.g. tail = [sam, jane, sally].

The empty list [ ] is a subset of all lists. Prolog uses a vertical bar to indicate the separation of the head and tail of a list.

e.g. [X | Y]

Example of recursion is list membership.

X is a member of list L if either

1. X is the head of L

or

2. X is a member of the tail of L.

```
member(X, [ X | Tail ]).
member(X, [ Head | Tail ]) if member(X, Tail).
```

Frames in Prolog

A frame consists of a name followed by an arbitary number of slots, each of which can support an arbitary number of facets [Cuadrado]. Each facet has an associated value. The value associated with a given facet can be an integer, string, list or an even more complex object. The most common facet is the value facet.

The facet refers to the actual value taken by the corresponding slot. Other commonly occuring facets include the default and if_needed facets.

The various functions that provide retrieval and maintenance facilities for this particular representation of frames in Prolog are described below.

A function is needed that will retrieve information from the slots in a frame. The predicate frame_get performs such a function. The predicate frame_put permits the installation of value in a given slot of a frame. Frame_remove is a predicate used for the removal of information from a frame. Occasionally, we not only need to install and remove values from some slot in a frame but also need to replace whatever value is in a slot with a new value. The predicate frame_replace does this. There are times when we need to deal with lists as the values of some slots. Often in these cases we build the lists incrementally. The predicate frame_append appends values to a list in a slot.

PEX (PACKAGING EXPERT)

PEX is an expert system that will select or design a suitable package for a silicon die. The expert system is implemented in Turbo Prolog. It utilizes both a forward chaining as well as backward chaining process. The backward chaining process is used for the selection of an existing package and reduces the search space. The forward chaining process enables the design of a new package. The mode of chaining describes the way in which the rules are activated, e.g. by matching working memory elements against one side and then processing the other side. The types of packages under consideration were the dip, soic, plcc, clcc, lccc and pga.

In the backward chaining process there is a goal-driven strategy based upon the principle of resolution. Resolution is the process of starting with an initial goal statement and resolve it with one of the hypotheses to create a new clause. This process continues until it reaches a point where the new clause can be conclusively answered. In Prolog terms, goals lead to sub-goals, and sub-goals lead to other sub-goals, etc. This process continues until the sub-goals are satisfied [Clocksin]

PEX is an user friendly, menu driven system. The expert system is part of a bigger system that handles other aspects of the packaging technology. The different modules of the system are:

- Expert system
- CAD
- Materials database
- Algorithms

Each of these modules can be accessed separately. In the final phase of the SRC project all these modules will be capable of interacting with each other. All the above modules are accessible through a user interface.

PEX STRUCTURE

PEX used a frame based approach in representing the knowledge. It utilized the source code for frame manipulation from BYTEnet listing at (617) 861-9764. The expert system can be accessed through the user interface and the logo for the epxert system is displayed. The information specific to the company is held in technology files. The expert system has the following menu options:

- INPUT
- RUN
- SHOW
- EXPLAIN
- SAVE
- EXIT

A selection can be made by moving the cursor to the appropriate mode and hitting the *F10* key.

Input Mode

This mode allows the user to input details of the package requirements. The screen contains prompts for the questions in the form of pull down menus. A suitable selection can be made by hitting the ***enter*** key. For numerical input data, the user is expected to key in a reasonable value for the parameter. The input parameters pertain to the following issues:

- Number of I/Os
- Hermeticity requirements
- Thermal dissipation
- Physical dimensions of the die
- Technology base
- Desired material
- Type of application

Based upon the responses to the above input, the expert system can figure out a package design.

Run Mode

This mode is used to process the data obtained from the input mode. The run mode is activated by selecting the *Run* option and hitting the *F10* key. The inference engine then figures out a suitable package (if possible) from the requested knowledge base.

Show Mode

This mode gives the user the final analysis of the design without providing any reasons for arriving at the package design. It gives all the pertinent information about the package like the cavity size, the physical dimensions, wire bond information, material, number of pins etc.

Explain Mode

This mode gives a lot more information than the show mode. It traces the logic structure of the expert system and gives information as to why it took a particular decision. All the rules that were fired are included in this mode.

Save Mode

This mode enables the user to store the design of a particular package that was derived from the expert system. This can be used later without having to run the software again.

Exit Mode

This mode brings the user out of the expert system to the user interface.

BIBLIOGRAPHY

BOOKS

Jackson, Peter, Introduction to Expert Systems.
Addison Wesley Publishing Company, 1986.

Clocksin, W. F., and Mellish, C. S., Programming in Prolog, 2nd ed.
New York: Springer-Verlag, 1981.

Winston, Patric Henry, Artificial Intelligence, 2nd ed.
Reading, Massachusetts: Addison Wesley Publishing Company, 1984.

________, Turbo Prolog, Borland International, 1986.

________, JEDEC Registered and Standard Outlines for Semiconductor Devices,
Publication No. 95.
Washington D. C. : Electronics Industries Association, Feburary 1988.

ARTICLES

Cuadrado, J. L., and Cuadrado, C. Y., "AI in Computer Vision" BYTE, January 1986, pp 237- 258.

________, "Packaging Technology Trends", IEEE 1982, Chap VI, pp 27 - 38.

Nelson, J. A., "VLSI Package Design Considerations" IEEE 1981, pp 320 - 325.

Bowlby, R., "The DIP May Take its Final Blow" IEEE Spectrum, June 1985, pp 37 - 42.

Marcus, S., Stout, J., and McDermott, J., "VT: An Expert Elevator Designer That Uses

Knowledge-Based Backtracking" AI Magazine, Vol 8, No. 4, Winter 1987, pp 41 - 58.

Denning, P. J., "Towards a Science of Expert Systems" IEEE Expert, Summer 1986, pp 80 - 83.

Dreyfus, H., and Dreyfus, S., "Why Expert Systems Do Not Exhibit Expertise" IEEE Expert, Summer 1986, pp 86 - 90.

Chandrasekaran, B., "Generic Tasks in Knowledge Based Reasoning; High-Level Building Blocks for Expert Systems Design" IEEE Expert, Fall 1986, pp 23 - 30.

Williams, C., "Expert Systems, Knowledge Engineering and AI tools - An Overview" IEEE Expert, Winter 1986, pp 66 - 70.

Freedman, D. H., "Expert System Vendors Aim for Simpler, Lower-Cost Packages" High Technology, April 1987, pp 26.

Davis, D. B., " Artificial Intelligence Goes to Work" High Technology, April 1987, pp 16 - 24.

Teschler, L., "What You Need To Know About Expert Systems" Machine Design, May 21, 1987, pp 85 - 91.

UNPUBLISHED MATERIAL

Advanced Micro Devices Package Design Guidelines.

Kyocera Package Design Guidelines.

APPENDIX A

SHOW MODE e.g.1

--------------------- DESIGN ---------------------

Type of package : pga
Type of material : ceramic
Number of pins on chip : 256
Type of circuit being packaged : microprocessor
Hermeticity requirements : Hermetic
Number of pins on package : 256
Die length : 300 mils
Die width : 300 mils
Maximum die length for package : 800 mils
Maximum die width for package : 770 mils
Maximum die attach clearance X : 40 mils
Minimum die attach clearacne X : 20 mils
Maximum die attach clearance Y : 60 mils
Minimum die attach clearance Y : 35 mils
Body length : 2060 mils
Body width : 2060 mils
Cavity length : 870 mils
Cavity width : 870 mils

Modifications and additions to above parameters
These are the design rules for a ceramic pga
Packaging Materials

Ceramic Material Properties

ColorDark

Alumina content90% min

Flexural strength 3000 Kg/cm2

Dielectric strength10 Kv/mm

Specific gravity3.8

Coeff of linear expansion7.1 x 10^-6 /C

Thermal conductivity (25 C)0.03 cal/ cm sec C

Dielectric constant (1 MHz @ 25 C)10

Ceramic Material Properties

LS-0113T187

Coeff of thermal expansion (30 - 250 C)6.4 x 10^-66.8 x 10^-6

Specific gravityg/cc6.855.60

Transition point C315308

Softening point C400342

Sealing temperature C450430

Dielectric loss tangent0.00850.0030

Dielectric constant (1MHz @ 25 C)33.012.80

Volume resistivity

at 25 C log(cm)11.911.9

at 250 C log(cm) 9.49.4

Thermal Conductivity (25 C)0.781.0

Alpha particle emission count/hr cm24.5 max0.8 max

External Package Features

Pin 1 ID : Dot on top side, addition of dummy pin on bottom side

Pin requirements:

Pin diameter20/17 mils

Stand off length55/45 mils

Pin Length (total)180 +/- 10 mils

Row to row pin spacing100 mils basic

SHOW MODE e.g. 2

--------------------- DESIGN ---------------------

Type of package : dip
Type of material : plastic
Number of pins on chip : 20
Type of circuit being packaged : memory
Hermeticity requirements : Non Hermetic
Number of pins on package : 20
Die length : 20 mils
Die width : 20 mils
Maximum die length for package : 256 mils
Maximum die width for package : 128 mils
Maximum die attach clearance X : 40 mils
Minimum die attach clearacne X : 20 mils
Maximum die attach clearance Y : 60 mils
Minimum die attach clearance Y : 35 mils
Body length : 954 mils
Body width : 288 mils
Cavity length : 574 mils
Cavity width : 168 mils

Modifications and additions to above parameters
Fixed data for plastic dips:-

Frame material : Tamac 5 or Olin 194
Properties of Tamac 5
Temper : hard
Tensile strength KPSI : 65 - 74
Yield strength KPSI : 63 - 74
Elongation : 4 - 8 %

Properties of Olin 194

Temper : Spring (RA)

Tensile strength KPSI : 70 - 74

Yield strength KPSI : 63 - 74

Elongation : 4 - 7 %

Molding compound

Vendor/Compound : Sumitomo EME 6300

Tg : 165 C

Flexural strength : 1200 Kg/mm

Wire type : 99.99 %

Tensile : 14 - 18g

Elongation : 3 - 7%

Typical properties of the molding compound EME 6300 are

Thermal Expansion : 2.0 x 10^-5 1/C max

Thermal Conductivity : 14 x 10^-4 cal/cm sec C min

Glass Transition Temp. : 155 C min.

Heat Deflection Temp. : 200 C min

Water Absorption after 24 hours boiling : 0.5 % max

Specific Gravity : 1.76 - 1.82

Extractable Cl- : 25 ppm max

Extractable Br- : 25 ppm max

Extractable Na+ : 10 ppm max

Hydrolyzable Chloride : 150 ppm max

Flexural Strength : **10 Kg min**

Volume Resistivity (Room Temp) : 5 x 10^14 ohm-cm min

Volume Resistivity (150 C) : 2 x 10^13 ohm-cm min

Dielectric Constant : 4.7 max

Flammability : V.O

Die Attach Epoxy Properties

Thermal Conductivity : 17 x 10^-2 watt/cm C min

Glass transition temp : 70 C min

Thermal Expansion : 65 x 10^-6 in/in C max

Extractable Cl- : 40 ppm

Extractable Na+ : 25 ppm max

Extractable K+ : 20 ppm max

Volume Resistivity : 0.001 ohm-cm max

Viscosity : 40 K - 60 K cps

Plating options : 200 u min 60/40 Sn/Pb

35 x 8 mils depth round hole centered 50 mils from edges adjacent to pin 1

--------------------- DESIGN ---------------------

Type of package : plcc

Type of material : plastic

Number of pins on chip : 20

Type of circuit being packaged : memory

Hermeticity requirements : Non Hermetic

Number of pins on package : 20

Die length : 20 mils

Die width : 20 mils

Maximum die length for package : 210 mils

Maximum die width for package : 210 mils

Maximum die attach clearance X : 40 mils

Minimum die attach clearacne X : 10 mils

Maximum die attach clearance Y : 40 mils

Minimum die attach clearance Y : 10 mils

Body length : 352 mils

Body width : 352 mils

Cavity length : 230 mils

Cavity width : 230 mils

Modifications and additions to above parameters

Technology base has to be upgraded.

---------------------- DESIGN ----------------------

Type of package : soic

Type of material : plastic

Number of pins on chip : 20

Type of circuit being packaged : memory

Hermeticity requirements : Non Hermetic

Number of pins on package : 20

Die length : 20 mils

Die width : 20 mils

Maximum die length for package : 180 mils

Maximum die width for package : 150 mils

Maximum die attach clearance X : 40 mils

Minimum die attach clearacne X : 10 mils

Maximum die attach clearance Y : 40 mils

Minimum die attach clearance Y : 10 mils

Body length : 508 mils

Body width : 294 mils

Cavity length : 200 mils

Cavity width : 170 mils

Modifications and additions to above parameters

Technology base has to be upgraded.

---------------------- DESIGN ----------------------

Type of package : clcc

Type of material : ceramic

Number of pins on chip : 20

Type of circuit being packaged : memory

Hermeticity requirements : Non Hermetic

Number of pins on package : 28

Die length : 20 mils

Die width : 20 mils

Maximum die length for package : 304 mils

Maximum die width for package : 334 mils

Maximum die attach clearance X : 40 mils

Minimum die attach clearacne X : 20 mils

Maximum die attach clearance Y : 60 mils

Minimum die attach clearance Y : 35 mils

Body length : 474 mils

Body width : 474 mils

Cavity length : 374 mils

Cavity width : 374 mils

Modifications and additions to above parameters

Technology base has to be upgraded.

--------------------- DESIGN ---------------------

Type of package : lccc

Type of material : ceramic

Number of pins on chip : 20

Type of circuit being packaged : memory

Hermeticity requirements : Non Hermetic

Number of pins on package : 20

Die length : 20 mils

Die width : 20 mils

Maximum die length for package : 120 mils

Maximum die width for package : 150 mils

Maximum die attach clearance X : 40 mils

Minimum die attach clearacne X : 20 mils

Maximum die attach clearance Y : 60 mils

Minimum die attach clearance Y : 35 mils

Body length : 350 mils

Body width : 350 mils

Cavity length : 190 mils

Cavity width : 190 mils

Modifications and additions to above parameters

Technology base has to be upgraded.

APPENDIX B

EXPLAIN MODE e.g. 1

LIST OF POSSIBLE PACKAGES: ["pga"]

Trying pga ...

Because the initial input was the following:

NUMBER OF LEADS on chip	: 256
HERMETICITY requirement	: Hermetic
TYPE OF CIRCUIT on chip	: microprocessor
DIE LENGTH x DIE WIDTH (mils)	: 300 x 300

The list of possible packages that can encapsulate this chip
are ["pga"]

Because of the NUMBER_OF_LEADS on the chip (256), The type of
package first recommended is the pga.

The DIE SIZE is 300 x 300 and the maximum DIE SIZE that this package
can encapsulate is 800 x 770. Comparison of the DIE SIZE
to the maximum DIE SIZE is OKAY.

These are the design rules for a ceramic pga
Packaging Materials

Ceramic Material Properties

ColorDark

Alumina content90% min

Flexural strength 3000 Kg/cm2

Dielectric strength10 Kv/mm

Specific gravity3.8

Coeff of linear expansion7.1 x 10^-6 /C

Thermal conductivity (25 C)0.03 cal/ cm sec C

Dielectric constant (1 MHz @ 25 C)10

Ceramic Material Properties

LS-0113T187

Coeff of thermal expansion (30 - 250 C)6.4 x 10^-66.8 x 10^-6

Specific gravityg/cc6.855.60

Transition point C315308

Softening point C400342

Sealing temperature C450430

Dielectric loss tangent0.00850.0030

Dielectric constant (1MHz @ 25 C)33.012.80

Volume resistivity

at 25 C log(cm)11.911.9

at 250 C log(cm) 9.49.4

Thermal Conductivity (25 C)0.781.0

Alpha particle emission count/hr cm24.5 max0.8 max

External Package Features

Pin 1 ID : Dot on top side, addition of dummy pin on bottom side

Pin requirements:

Pin diameter20/17 mils

Stand off length55/45 mils

Pin Length (total)180 +/- 10 mils

Row to row pin spacing100 mils basic

EXPLAIN MODE e.g. 2

LIST OF POSSIBLE PACKAGES: ["lccc","clcc","soic","plcc","dip"]

Trying dip ...

Because the initial input was the following:

NUMBER OF LEADS on chip : 20
HERMETICITY requirement : Non Hermetic
TYPE OF CIRCUIT on chip : memory
DIE LENGTH x DIE WIDTH (mils) : 20 x 20

The list of possible packages that can encapsulate this chip are ["lccc","clcc","soic","plcc","dip"]

Because of the NUMBER_OF_LEADS on the chip (20), The type of package first recommended is the dip.

The DIE SIZE is 20 x 20 and the maximum DIE SIZE that this package can encapsulate is 256 x 128. Comparison of the DIE SIZE to the maximum DIE SIZE is OKAY.

The properties of a plastic dip

Fixed data for plastic dips:-

Frame material : Tamac 5 or Olin 194

Properties of Tamac 5

Temper : hard

Tensile strength KPSI : 65 - 74

Yield strength KPSI : 63 - 74

Elongation : 4 - 8 %

Properties of Olin 194

Temper : Spring (RA)

Tensile strength KPSI : 70 - 74

Yield strength KPSI : 63 - 74

Elongation : 4 - 7 %

Molding compound

Vendor/Compound : Sumitomo EME 6300

Tg : 165 C

Flexural strength : 1200 Kg/mm

Wire type : 99.99 %

Tensile : 14 - 18g

Elongation : 3 - 7%

Typical properties of the molding compound EME 6300 are

Thermal Expansion : 2.0 x 10^-5 1/C max

Thermal Conductivity : 14 x 10^-4 cal/cm sec C min

Glass Transition Temp. : 155 C min.

Heat Deflection Temp. : 200 C min

Water Absorption after 24 hours boiling : 0.5 % max

Specific Gravity : 1.76 - 1.82

Extractable Cl- : **25 ppm max**

Extractable Br- : **25 ppm max**

Extractable Na+ : **10 ppm max**

Hydrolyzable Chloride : **150 ppm max**

Flexural Strength : **10 Kg min**

Volume Resistivity (Room Temp) : 5 x 10^14 ohm-cm min

Volume Resistivity (150 C) : 2 x 10^13 ohm-cm min

Dielectric Constant : 4.7 max

Flammability : V.O

Die Attach Epoxy Properties

Thermal Conductivity : 17 x 10^-2 watt/cm C min

Glass transition temp : 70 C min

Thermal Expansion : 65 x 10^-6 in/in C max

Extractable Cl- : 40 ppm

Extractable Na+ : 25 ppm max

Extractable K+ : 20 ppm max

Volume Resistivity : 0.001 ohm-cm max

Viscosity : 40 K - 60 K cps

Plating options : 200 u min 60/40 Sn/Pb

35 x 8 mils depth round hole centered 50 mils from edges adjacent to pin 1

Trying plcc ...

Because the initial input was the following:

NUMBER OF LEADS on chip : 20

HERMETICITY requirement : Non Hermetic

TYPE OF CIRCUIT on chip : memory

DIE LENGTH x DIE WIDTH (mils) : 20 x 20

The list of possible packages that can encapsulate this chip
are ["lccc","clcc","soic","plcc","dip"]

Because of the NUMBER_OF_LEADS on the chip (20), The type of
package recommended is the plcc.

The DIE SIZE is 20 x 20 and the maximum DIE SIZE that this package
can encapsulate is 210 x 210. Comparison of the DIE SIZE
to the maximum DIE SIZE is OKAY.
Technology base has to be upgraded.

Trying soic ...

Because the initial input was the following:

NUMBER OF LEADS on chip : 20
HERMETICITY requirement : Non Hermetic
TYPE OF CIRCUIT on chip : memory
DIE LENGTH x DIE WIDTH (mils) : 20 x 20

The list of possible packages that can encapsulate this chip
are ["lccc","clcc","soic","plcc","dip"]

Because of the NUMBER_OF_LEADS on the chip (20), The type of
package first recommended is the soic.

The DIE SIZE is 20 x 20 and the maximum DIE SIZE that this package
can encapsulate is 180 x 150. Comparison of the DIE SIZE
to the maximum DIE SIZE is OKAY.

Technology base has to be upgraded.

Trying clcc ...

Because the initial input was the following:

NUMBER OF LEADS on chip : 20
HERMETICITY requirement : Non Hermetic
TYPE OF CIRCUIT on chip : memory
DIE LENGTH x DIE WIDTH (mils) : 20 x 20

The list of possible packages that can encapsulate this chip are ["lccc","clcc","soic","plcc","dip"]

Because of the NUMBER_OF_LEADS on the chip (20), The type of package first recommended is the clcc.

The DIE SIZE is 20 x 20 and the maximum DIE SIZE that this package can encapsulate is 304 x 334. Comparison of the DIE SIZE to the maximum DIE SIZE is OKAY.

Technology base has to be upgraded.

Trying lccc ...

Because the initial input was the following:

NUMBER OF LEADS on chip : 20
HERMETICITY requirement : Non Hermetic
TYPE OF CIRCUIT on chip : memory
DIE LENGTH x DIE WIDTH (mils) : 20 x 20

The list of possible packages that can encapsulate this chip
are ["lccc","clcc","soic","plcc","dip"]

Because of the NUMBER_OF_LEADS on the chip (20), The type of
package first recommended is the lccc.

The DIE SIZE is 20 x 20 and the maximum DIE SIZE that this package
can encapsulate is 120 x 150. Comparison of the DIE SIZE
to the maximum DIE SIZE is OKAY.
Technology base has to be upgraded.

VITA

Ravi Kulasekaran, son of R. Kulasekaran and Ramani Kulasekaran, was born in Madras, India, March 25, 1964. In 1986, he received a Bachelor of Engineering in Mechanical Engineering from Delhi College of Engineering, Delhi University. He then joined the Manufacturing Systems Engineering program at Lehigh University for a Master of Science in Industrial Engineering. He was involved with Ben Franklin projects as a Project Assistant. Later, he became a research assistant with the Computer Science department at Lehigh University and worked on the development of an expert system for the Semiconductor Research Corporation. After graduating, he will be involved in the development and manufacture of integrated circuit chip packages as a packaging engineer.