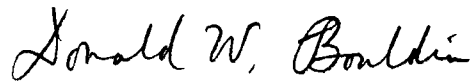


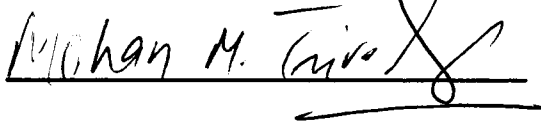
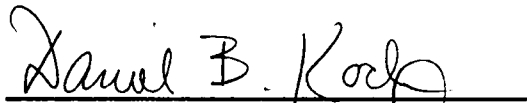
To the Graduate Council:

I am submitting herewith a thesis written by Subhomoy Chattopadhyay entitled "AUTOMATED COMPONENT PLACEMENT IN MULTICHIP MODULES USING MULTIPLE CRITERIA." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.



Prof. Donald W. Bouldin, Major Professor

We have read this thesis
and recommend its acceptance:



Accepted for the Council:



Associate Vice Chancellor
and Dean of The Graduate School

**AUTOMATED COMPONENT PLACEMENT IN
MULTICHIP MODULES USING MULTIPLE
CRITERIA**

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Subhomoy Chattopadhyay

August 1995

ACKNOWLEDGEMENT

I would like to express my special thanks to my major professor Prof. Don Bouldin for all his help and encouragement throughout the course of the work that is described in this thesis. I would also like to sincerely thank Prof. Daniel B. Koch and Prof. M.M. Trivedi for serving on my thesis committee and for the many valuable suggestions they gave.

I would also like to express my thanks and gratitude to Prof. Dinesh Bhatia and his students at University of Cincinnati, Ohio for giving us a part of the software that has been used in this thesis. I sincerely acknowledge their help. I would also like to thank the guys in our lab Kanagaraj and Barry Tolnas for their help, encouragement and support when it was needed the most.

However this work would never have been possible without the love, affection, support and encouragement of my parents. Their love and sacrifices have made me a much better person today. My special thanks are also due to my beloved wife Baishali, for all her love, support and company during the course of work for this thesis. She made my stay in USA a much more memorable one.

ABSTRACT

In this thesis we study some of the approaches to placement of components (dies) in multichip modules. A simulated annealing based electrical/thermal placement tool has been developed that runs in a PVM (Parallel Virtual Machine) environment. *Placement* and *routing* constitute the main tasks in physical design. Placement of multiple dies on an MCM substrate is a non-trivial task in which multiple criteria need to be considered simultaneously. An ideal auto-placement tool should result in the smallest layout while conforming to electrical and thermal requirements. This thesis describes in detail the software that has been developed to perform placement of components (dies) in an MCM with thermal criteria taken into consideration simultaneously along with the objective of total netlength (delay and area) minimization. By doing a simultaneous auto-placement considering both thermal and netlength objectives we see that the final netlength after optimization is higher on average by 10 percent compared to the case when only netlength minimization is done; the heat penalty however is much lower. Executing the applications in a parallel mode results in speedup by a factor of *eight* for a 37 die, 7118 net MCM example when we use 11 Sparcstations. Our approach reduces the system design time while achieving the same quality of placement. By executing the optimization in parallel (PVM) the runtime can be substantially reduced (factor of eight). In addition our approach also allows a quick exploration of the design space to obtain a near optimal solution and a more uniform thermal distribution.

TABLE OF CONTENTS

CHAPTER		PAGE
1. INTRODUCTION TO PACKAGING AND MULTICHIP		
MODULES		1
1.1	VLSI and MCM Packaging	2
1.2	Advantages of using Multichip Packaging	5
1.3	Introduction to Multichip Modules (MCMs)	7
1.4	Challenges Faced by MCM Designers of Today	9
1.4.1	Packaging Efficiency	9
1.4.2	Performance Model of an MCM-based System	10
1.5	Chapter Contents	11
2. RELATED WORK IN THE AREA OF PLACEMENT FOR		
MULTICHIP MODULES		12
2.1	Introduction to the Physical Design of Multichip Modules	12
2.2	Physical Design Flow for MCMs	13
2.2.1	Steps in the Physical Design Process	15
2.3	Motivation for the Research in this Thesis	17
2.4	Brief Overview of the Thesis	18
2.5	Related Research in MCM Physical Design - Algorithms and Ap- proaches	19

CHAPTER	PAGE
3. ELECTRICAL/THERMAL DRIVEN PLACEMENT OF DIES ON THE MCM SUBSTRATE	20
3.1 Introduction	20
3.2 Need for Consideration of Thermal Constraints in MCM	21
3.2.1 Approach to the Thermal Management problem	22
3.3 Problem Formulation for Placement	23
3.4 Placement as an NP-complete Problem	26
3.4.1 Algorithmic Approaches to NP-hard Problems	27
3.5 Simulated Annealing Algorithm	27
3.6 Cost Function for Optimization	30
3.6.1 Calculation of the Cost Function	31
3.7 Moves Used to Attain an Optimal Solution	31
3.8 Electrical/Thermal Placement Results	33
3.9 Multi-Objective Optimization	37
3.10 Improvement in System Design Time	39
3.11 Analysis of the Results	42
3.12 Conclusions	48
4. COMPLETE MCM PHYSICAL DESIGN USING MENTOR GRAPHICS MCM STATION	49
4.1 Introduction	49
4.2 The Design Objects Available in Mentor Graphics MCM Station .	52
4.3 LAYOUT Tool in MCM Station	53

CHAPTER	PAGE
4.3.1 MCM Placement	53
4.3.2 Interactive Placement	54
4.3.3 Thermal Analysis After Placement with AutoTherm . . .	56
4.3.4 Auto Routing using the Layout Tool	58
4.4 MCM Physical Design Using Design Kits	63
4.5 Conclusions	63
5. PARALLEL IMPLEMENTATION OF THE SIMULATED AN- NEALING BASED PLACEMENT USING PVM	65
5.1 Introduction to Parallel Virtual Machine (PVM) and XPVM . . .	65
5.2 Simulated Annealing Algorithm in a PVM Environment	66
5.2.1 Advantages of using a PVM with SA Algorithm	66
5.2.2 Implementation of a PVM	67
5.2.3 Parallel Simulated Annealing Algorithm for MCM Placement	69
5.2.4 Implementation of the SA Algorithm	70
5.2.5 PVM Call Constructs in the Simulated Annealing Algorithm	70
5.3 Performance Comparison when using a PVM	72
5.4 Comments and Conclusions	77
6. SUMMARY, CONCLUSION AND FUTURE WORK	80
6.1 Summary and Conclusion	80
6.2 Future Work	81
BIBLIOGRAPHY	83

CHAPTER	PAGE
APPENDIX	86
A. AUTOROUTER ROUTING REPORT FROM MCM STA- TION FOR VIPER DESIGN	87
VITA	99

LIST OF FIGURES

FIGURE	PAGE
1.1 <i>Conventional Levels of Packaging</i>	3
1.2 <i>Area Savings when using PCB and MCM Packaging</i>	4
2.1 <i>MCM Physical Design Flow</i>	14
3.1 <i>Placement of Dies on the MCM Substrate [42]</i>	25
3.2 <i>Estimation of Netlength using Half Perimeter Method</i>	32
3.3 <i>Tradeoff Analysis Curve for the Viper Placement using MODA . .</i>	38
3.4 <i>Selection of the best candidate solution using MODA</i>	40
3.5 <i>Improvement in System Design Time</i>	41
3.6 <i>MAGIC Layout for Viper on the MCM Substrate without Thermal Considerations</i>	44
3.7 <i>MAGIC Layout for Viper on the MCM Substrate with Thermal Considerations</i>	45
3.8 <i>MAGIC Layout for mcc2 on the MCM Substrate without Thermal Considerations</i>	46
3.9 <i>MAGIC Layout for mcc2 on the MCM Substrate with Thermal Considerations</i>	47
4.1 <i>MCM Physical Design - Flow using Mentor Graphics MCM Station</i>	51
4.2 <i>Rats Nest Display for the Viper Placement</i>	55
4.3 <i>Thermal Model used in AUTOTHERM</i>	57

FIGURE	PAGE
4.4 <i>Junction temperatures for Viper design</i>	59
4.5 <i>LAYOUT, Autorouter Program Flow</i>	60
4.6 <i>Routed Viper Design</i>	62
5.1 <i>PVM Implementation Scheme</i>	68
5.2 <i>XPVM GUI with the Different Menus</i>	73
5.3 <i>PVM Spawning Process Running on Different Machines</i>	74
5.4 <i>Speedup Obtained for the Mcc2 Design (37 dies)</i>	75
5.5 <i>Speedup Obtained for the Viper Design (10 dies)</i>	76

LIST OF TABLES

TABLE		PAGE
1.1	<i>Performance and Cost Factors Affecting Packaging Decisions . . .</i>	6
1.2	<i>Application Scenarios of Multichip Modules</i>	8
3.1	<i>Design Data for the Example Problems</i>	33
3.2	<i>Results for Placement with the Thermal and Netlength Criteria . .</i>	34
3.3	<i>Results for Placement with only the Netlength Criteria</i>	35
3.4	<i>Comparison of results With and Without the Thermal Criteria (av-</i> <i>erage netlength and heat penalty)</i>	36
5.1	<i>Run Times and Speedup for MCM Placement of Mcc2 in a PVM .</i>	78
5.2	<i>Run Times and Speedup for MCM Placement of a 10 die, Viper</i> <i>Microprocessor in a PVM</i>	79

CHAPTER 1

INTRODUCTION TO PACKAGING AND MULTICHIP MODULES

The rapid development of VLSI technology has led to an exponential improvement in the performance and density of VLSI circuits and systems. This growth has been facilitated by the rapid improvement in process technology, devices, circuit design and physical design. As a result there has been a dramatic improvement in the size, cost and speed of VLSI systems [15]. Some of the available chips today can execute about 400 - 500 million instructions per second. However not all the components in the VLSI system have kept up with the rapid improvements in integrated circuit technology. The demand for smaller area and volume has forced designers to look at the packaging aspects in a way that has never been done before. Until a few years back, advanced packaging techniques were used only in designs for high performance computing applications. In the early 1980s IBM came up with TCM (thermal conduction modules) for its mainframes. In recent times multichip modules (MCMs) have been developed as a solution to this packaging bottleneck for a wide variety of applications. The performance of many systems of tomorrow will be limited because of the packaging bottleneck resulting in a more widespread use of MCM technology in the future.

1.1 VLSI and MCM Packaging

Figure 1.1 shows the different conventional levels of packaging [15]. On the first level bare dies are put into a package called the single chip package (SCP). Several of these single chip packages (also known as chips) are then put together on a printed circuit board (PCB). Then several of these PCBs are mounted on a backplane. Each successive level of packaging has larger feature size and hence occupies a larger volume and results in a reduction in packaging density. This decrease in packaging density, while going through successive levels of packaging is a challenge to IC and packaging designers. On a typical PCB with several VLSI chips the *silicon efficiency* (ratio of the total area occupied by silicon to the total board area between the cases when bare dies are put on the PCB and when packaged dies are put) is typically around five [15]. This reduction in feature size and hence the volume shows great promise for the development of multichip modules. Figure 1.2 shows the comparison between the cases when packaged dies are put on the PCB and when bare dies are put on a substrate. The silicon dies are shown by the dark rectangles. The area savings obtained by the second configuration is the key behind the concept of *multichip modules*. The area savings can be an order of magnitude or even a little more. Another big plus in favor of multichip modules is the reduction in the signal delays. In a conventionally packaged system the signal has to travel through the I/O pads in the SCP and through the gold wires connecting the different single chip packages on the PCB to reach another die.

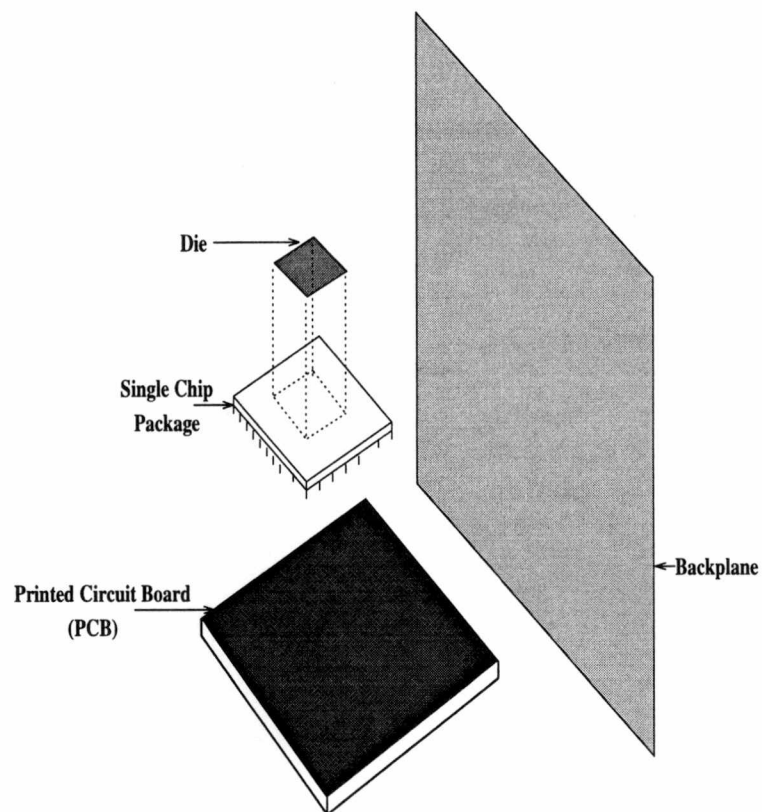
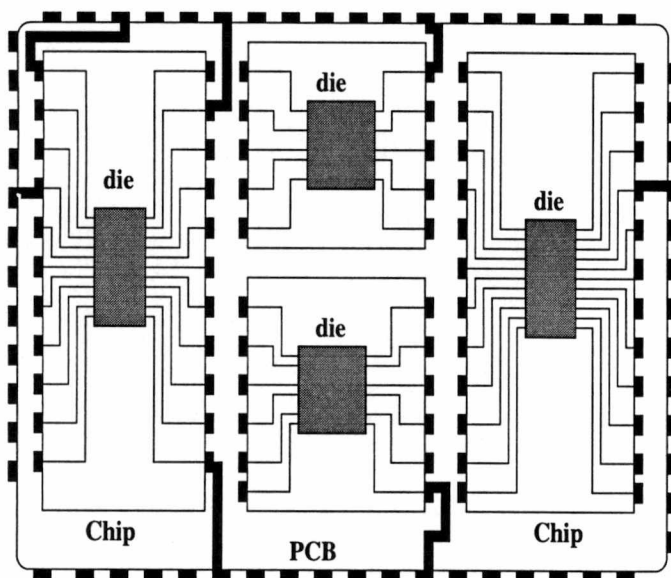
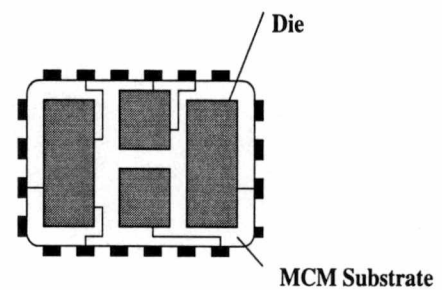


Figure 1.1: *Conventional Levels of Packaging*



(a) PCB Implementation



(b) MCM Implementation

Figure 1.2: Area Savings when using PCB and MCM Packaging

In an MCM the signals have to travel through the interconnect material. The system performance can be greatly enhanced by reducing the delay due to signal transmission across packages because the reduction in signal delays results in a decrease in system cycle time and hence increases the system operating frequency.

1.2 Advantages of using Multichip Packaging

During the system design process (especially for a large design) the system designers need to determine the organization of the system, packaging style to be used, the packaging mixture to be used at each level of hierarchy and the partitioning of the system functions to different chips that will comprise the system [9]. The term *organization* here means the block diagram of the whole design - what functional blocks make up the whole design and how are they interconnected. The term *packaging style* refers to the type of package to be used (whether MCM or SCP and if MCM what type, etc.). The term *partitioning* refers to what functions are assigned to which chips and how the chips are distributed among the different packages in the system. As part of the partitioning process we also include the actual physical design. Packaging and partitioning decisions are made by an early analysis tool. The requirements and goals of the system are expressed in terms of performance and cost. The performance factors are given in Table 1.1.

Table 1.1: *Performance and Cost Factors Affecting Packaging Decisions*

FACTORS AFFECTING PERFORMANCE
1. Volume and weight of the package 2. Interconnection capacity within each interconnection level in the design 3. Connection capacity between interconnection levels 4. Simultaneous switching noise and crosstalk and other noise 5. Power dissipation in the ICs 6. External and internal thermal resistance and temperature distribution
FACTORS AFFECTING SYSTEM COST
1. Cost of production: a. NRE (Non recurring engineering) cost b. Manufacturability cost (impact of testing and yield) 2. Post production costs (mainly maintenance and cost of replacing parts) 3. Design and prototyping costs and time to market

1.3 Introduction to Multichip Modules (MCMs)

A multichip module (MCM) is essentially a circuit consisting of many bare dies put together on a substrate in a single package. MCM technology holds multiple bare dies on a single chip package to attain the same functionality as a number of single chip packages (SCP's) put together on a printed circuit board (PCB).

The main advantages of multichip modules are:

- The board layout size can be reduced often by a factor of 10
- Signal propagation delays among the chips can be reduced by a factor of 10
- The solder connections can be reduced resulting in higher reliability
- Large number of I/Os possible with flip chip technology
- Increased number of interconnections are possible resulting in a dense designs.

MCMs reduce the interchip wiring delays that affect the system critical path and system timing. This is due to the fact that the circuit components that were originally placed in separate SCPs are now placed close to each other on the MCM substrate. This can be realized by looking at Figure 1.2. Some of the application scenarios of multichip modules are given in Table 1.2.

MCMs were first developed at IBM corporation for their high performance mainframe computers. IBM has been developing multilayer ceramic "Thermal Conduction Modules" (TCMs), housing close to a hundred bare dies. Until recent times the use of MCMs was limited to the very high performance computing scenario. However in recent years researchers and engineers in the area of VLSI have realized that packaging is the main bottleneck to the further decrease in

system cycle time. Current signal delay times of nanoseconds is expected to be in picoseconds by the beginning of 21st century. This difference in signal delay times is attributed to both on-chip circuitry and off-chip packaging technology. The development of communications and the need for better portable computing systems is another driving force in the development of packaging. Size, weight and power consumption reduction are key to the development of better portable computing and communications systems. These purposes are not served satisfac-

Table 1.2: *Application Scenarios of Multichip Modules*

<i>Applications</i>	<i>Special Characteristics</i>
<i>Computers</i>	<i>High performance computing Smaller space, higher reliability</i>
<i>Military</i>	<i>Less space and weight</i>
<i>Telecommunications</i>	<i>High performance computing</i>
<i>Consumer Electronics</i>	<i>Reliability, less space and weight</i>
<i>Medical</i>	<i>Less space and weight</i>
<i>Automotive</i>	<i>Less space and lighter</i>

torily by conventional packaging systems. The maximum reported working clock frequency of 348 MHz was achieved with the help of MCMs at Hughes Aircraft Company [5]. IBM's Enterprise system 9000, ES 9000 is built around an MCM that houses 121 chips and consists of a ceramic package consisting of 63 layers.

1.4 Challenges Faced by MCM Designers of Today

MCM designers face a variety of challenges in justifying their designs using MCM packaging since the packaging plays a major role in determining the capabilities and limitations of the system. Various packaging options should be considered from the early stages of the design. *Design for Packagability* [4] attempts to integrate the packaging and the overall system design process. This concept involves taking the packaging into consideration during IC design to optimize the overall system performance. Some of the main challenges faced by designers is listed below.

- *Early Package Analysis*: Inclusion of the packaging issues in the overall design process needs decisions to be made about the overall system performance. To handle this extra complexity added to the design process it is necessary to use an *early analysis* tool to determine the system performance and the options available to the designer.
- *Physical Design*: MCM physical design (which includes placement and routing mainly) poses a new set of challenges to the VLSI design engineer because the existing VLSI CAD tools are not adequate for MCM design.

1.4.1 Packaging Efficiency

There is a great improvement in packaging efficiency which results by placing multiple dies on the MCM substrate close to each other and then packaging them

as one unit as an MCM package instead of a single chip package. The packaging efficiency P_{eff} is defined as,

$$P_{eff} = \frac{SiliconArea}{TotalPackageArea} \quad (1.1)$$

1.4.2 Performance Model of an MCM-based System

A good measure of the electrical performance of a system is the MIPS (millions of instructions per second). The MIPS figure of merit is defined as follows :

$$MIPS = \frac{1000}{T_c * CPI} \quad (1.2)$$

where T_c is the processor clock period in nanoseconds and CPI is the clock cycle per instruction for a set of application programs. Both of these quantities mentioned above are functions of the system instruction and data caches. A tradeoff study is necessary to determine the optimal cache size. For example for a GaAs microprocessor developed as a flip chip MCM the optimal cache size is 8K [1]. It has also been experimentally determined that both semiconductor technology and packaging influence the MIPS and contribute about 50% each [5]. Comparative results of MCM implementation over other packaging technologies are shown as a graph in [18].

1.5 Chapter Contents

This thesis is divided into six chapters, beginning with the introduction to VLSI and MCM packaging in general. Chapter 2 deals with the introduction to the physical design problem for multichip modules. We also describe the motivation to do an electrical/thermal placement of components (dies) in a MCM. Chapter 3 discusses the approach and software for the automatic electrical/thermal for dies in multichip modules using multiple criteria simultaneously. The fourth chapter has a discussion about the traditional MCM design approach using Mentor Graphics MCM station. Thermal analysis of a MCM design is done using Mentor Graphics *Autotherm* tool and the need for consideration of thermal criteria for MCM design is discussed. Chapter 5 discusses the parallel implementation of the automated placement algorithm using a PVM (Parallel Virtual Machine) environment. Chapter 6 deals with conclusions and future work.

CHAPTER 2

RELATED WORK IN THE AREA OF PLACEMENT FOR MULTICHIP MODULES

As mentioned in the last chapter one of the main reasons for using multichip modules instead of single chip packages is to improve the system performance in terms of speed and obtain a layout which is an order of magnitude less in area than that of a single chip package for the same design. This chapter outlines the work done in this thesis along with the motivation to do research in the area of placement of components and dies on the MCM substrate using multiple criteria simultaneously. A brief introduction is also provided in the initial part to explain the general design flow involved in the physical design of multichip modules.

2.1 Introduction to the Physical Design of Multichip Modules

MCM physical design consists of two operations: placement and routing. *Placement* refers to the positioning of unpackaged or bare dies on the MCM substrate such that one or more figures of merit like the overall netlength, timing delay and thermal distribution are optimized. *Routing* is the task of interconnecting the dies on the substrate such that one or more figures of merit are optimized. For routing typical optimization criteria include netlength, via, layer or crosstalk

minimization.

Placement and routing are mutually dependent but the size and complexity of practical problems generally dictate that each one of these problems be solved separately. Each operation is order-dependent in that the placing of one die or the routing of an interconnection leaves fewer options for the remaining dies to be placed or interconnections to be routed. Thus, those placements or interconnections which are considered to have a greater impact on the optimization criteria are performed first. This is done in most cases by using a weighting technique which gives preference to certain nets for example in the placement process to be placed first before the placement of other nets.

MCM physical design requires consideration of many issues which cannot be handled satisfactorily with computer-aided design tools developed for integrated circuits or printed circuit boards [18], [15], [4], [10]. Factors like the number of layers on the MCM substrate, the type of bonding and the type of substrate (ceramic, laminate or deposited) must be an integral part of the optimization procedure [14]. Considering the packaging issues early in the design cycle leads to better physical design and superior overall system performance [4].

2.2 Physical Design Flow for MCMs

Figure 2.1 illustrates the operations involved in MCM physical design. The top-level hardware description language (VHDL or Verilog) is synthesized into a structural netlist of library components and their interconnections. This large

netlist may then be partitioned into a set of different dies. Once the layouts of all the dies have been obtained, placement and routing of the MCM substrate are performed.

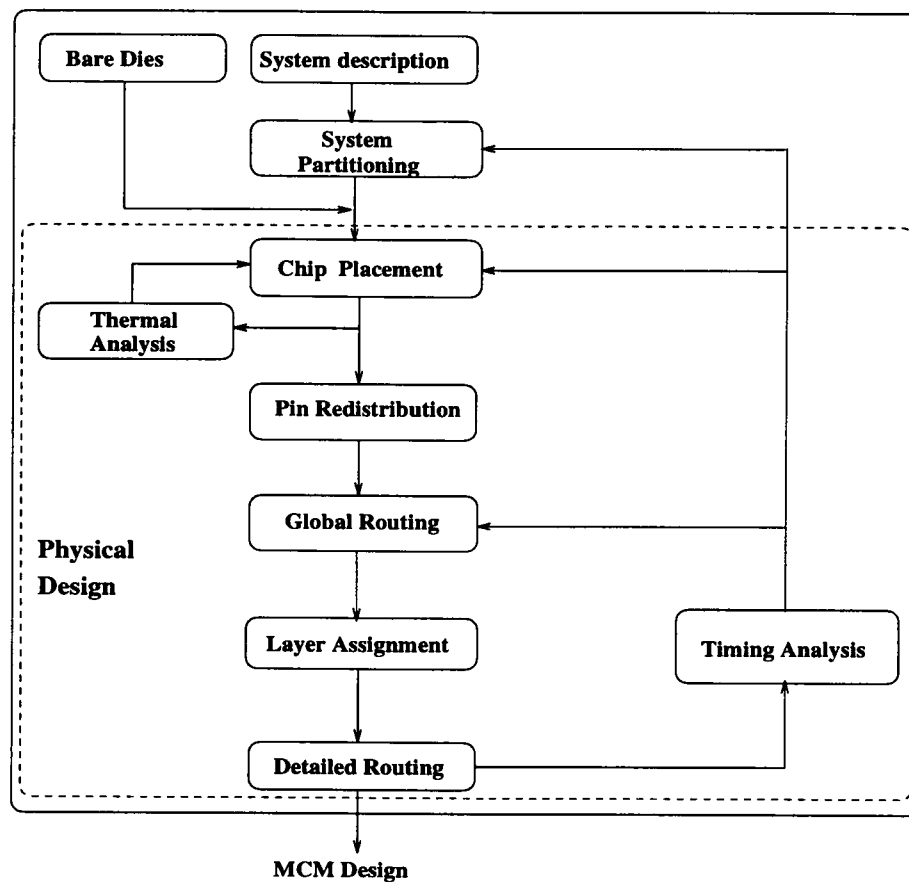


Figure 2.1: *MCM Physical Design Flow*

2.2.1 Steps in the Physical Design Process

Prior to the physical design process is the system partitioning step. Partitioning is the process of breaking up the functionality of a single large chip into multiple dies with the objective of maximizing performance and routability. This step is eliminated in cases where off the shelf bare dies are used. The next step in the physical design process is *chip placement*. MCM placement involves placement of dies on the MCM substrate such that the constraints of proper heat distribution, minimization of interchip delays, simultaneous switching noise and tolerant load distance is taken into consideration. However when the fabrication issues are considered additional constraints in the form of net separation constraint and via constraint come into the forefront since the fabrication of a very densely routed design may result in low fabrication yield [10].

The next step in the physical design process is *pin redistribution* [2]. In this step the I/O pins of the chips are uniformly distributed on the MCM substrate to increase the routability. The pin redistribution problem can be stated as follows: after the placement of the dies on the MCM substrate redistribute the pins using the pin redistribution layers such that the total number of pin redistribution layers, crosstalk and the maximum signal delay is minimized. Another objective of pin redistribution is to maximize the number of nets that can be routed in a planar fashion. Sometimes the pin redistribution and the placement steps are integrated into one step.

The final step in the physical design process is *routing* which usually consists

of *global routing* and *layer assignment* [15]. Layer assignment refers to the assignment of nets to different routing layers in the MCM under certain constraints. The MCM routing problem is quite different from the VLSI routing problem. In VLSI global routing the primary objective is to minimize the wirelength of the global routing trees since this reduces the chip area and net delays. MCM routing problem on the other hand is a three dimensional routing problem where the routing is not only done in the horizontal plane but also in the vertical direction in the multiple layers. The number of nets in the design, the substrate size, the number of chips mounted on the MCM substrate and the number of layers will continue to increase progressively causing the MCM routing problem to be of even greater importance. The main objectives of MCM routing can be the reduction of the number of routing layers, reduction or elimination of crosstalk which finally leads to an improvement of the overall system performance. Some of the performance related objectives are manufacturing constraints, netlength constraint, net-separation constraint and via constraint. The manufacturability constraint arises due to the concern about yield and cost of MCM fabrication. The fabrication related issues include types of bonding technology used, type of substrate used, use of staggered vias and maximization of net separation to reduce couplings [15].

2.3 Motivation for the Research in this Thesis

As we have mentioned earlier the placement of multiple dies on an MCM substrate is a non-trivial task in which multiple criteria need to be considered simultaneously. An ideal auto-placement tool for multichip modules should result in the smallest layout while conforming to electrical and thermal requirements. The use of MCMs reduces the volume required to house integrated circuits compared to those dies traditionally packaged individually and interconnected on a printed wiring board. Hence proper thermal management is a must in multichip modules because of the much smaller volume occupied by the MCM [7].

In MCMs proper heat management often becomes as vital as minimizing netlength (delay and area). Presently, the CAD tools used for MCM layout operate in isolation with the placement being performed based on netlength minimization without considering the thermal criteria for optimization. After a placement is done a thermal verification or optimization is done. The placement is then changed if necessary to satisfy the thermal constraints. The placement tool forces the dies to be close together while the thermal analysis tool often spreads them apart. Hence, it is desirable to consider both of these criteria simultaneously in order to obtain in less time the best solution which minimizes netlength yet still meets the thermal constraints.

2.4 Brief Overview of the Thesis

Thus the goal of this thesis work is to integrate these two criteria to provide a superior auto-placement while decreasing the system design time. We have used a simulated annealing based optimization algorithm to do the placement. However we realized that considering multiple criteria simultaneously increases the computational burden during the optimization process. The implementation of probabilistic optimization algorithms like simulated annealing require large computation time for design examples with large number of dies and nets. Most of the time in the placement process is spent in the calculation of the cost function rather than in the actual optimization. This makes for an ideal example of a parallelizable problem.

This thesis will also describe the software that has been developed to perform placement of components (dies) with the thermal criterion taken into consideration along with the objective of netlength minimization in a Parallel Virtual Machine (PVM) environment. The optimization algorithm based on simulated annealing has been modified partly to run in a PVM environment in parallel. The optimization algorithm works in a *master/slave* mode where the cost function calculations are done in a slave mode and the optimization is done in master mode. Chapter 5 describes this process in detail. In Chapter 4 the general methodology of design using state of the art MCM design EDA tool Mentor Graphics MCM station has been described.

2.5 Related Research in MCM Physical Design - Algorithms and Approaches

MCM placement is a difficult combinatorial problem as a good placement involves a tradeoff between different options with mutually conflicting constraints like netlength minimization and proper heat distribution constraint. A good placement should aid in improving the performance of the circuit by reducing the size and minimizing the signal delays. MCM routing is also a difficult combinatorial problem. A detailed description of the research in the area of MCM placement and routing is given in [16].

CHAPTER 3

ELECTRICAL/THERMAL DRIVEN PLACEMENT OF DIES ON THE MCM SUBSTRATE

3.1 Introduction

In the last chapter we have outlined the traditional approaches to MCM physical design where the electrical and the thermal constraints were separately considered for optimization to achieve the final placement. This chapter describes the software that has been developed to perform placement of dies on the MCM substrate with the thermal criteria taken into consideration along with the objective of netlength minimization (which achieves delay and area minimization). We have considered the examples where we have 6, 9, 10 and 37 partitioned dies for placement on the MCM substrate. The heart of this optimization routine is the simulated annealing algorithm which has been used to obtain a near optimal placement of dies with the multiple objective of *netlength minimization* and *uniform thermal temperature profile*. As mentioned earlier we will show that by doing a simultaneous auto-placement considering both the thermal and netlength criteria, the final netlength after optimization is higher than the case when they were considered separately. The heat penalty however goes down substantially.

The increase in netlength for moderately sized designs ie., designs with 6, 9 and 10 dies is minimal whereas it is a little more for large designs which have a large number of nets. A similar approach to MCM placement using only the netlength criteria was also developed by the author in [5].

3.2 Need for Consideration of Thermal Constraints in MCM

Supporting high heat fluxes while maintaining relatively low component temperatures is one of the major challenges facing MCM and PCB designers today [7]. The problem of thermal management exists in VLSI single chip modules, PCBs as well as MCMs. In order for an electronic system to operate properly the temperature at the transistor junctions must be maintained below a critical temperature. Electronic systems of today are becoming more compact and smaller through the use of advanced packaging techniques like MCMs. However the power dissipation in the devices is not decreasing and in many cases increasing with the higher interconnect density in MCMs. The thermal management problem in MCMs is much more critical because of the fact that an MCM generates much more heat per unit area compared to a single chip VLSI design (due to the much smaller feature size in MCMs). Removing this additional heat requires a consideration of the thermal issues early on in the physical design cycle. When a large number of dies are placed close to each other on the MCM substrate, as may be the case with MCNC benchmark *mcc2* which has 37 partitioned dies for a Honeywell supercomputer, the heat generated on the substrate can be sub-

stantial. If two dies that generate large quantities of heat are placed close to each other an uneven thermal distribution results. High temperatures increase the thermal shear stresses due to the thermal coefficient of expansion mismatch resulting in decreased noise margin, system reliability and MTTF (mean time to failure). The smaller area in MCMs decreases the effective heat dissipation surface area. Also MCM designers are faced with the challenge of management of power dissipation densities that are increasing fast nowadays. MCM designers can no longer ignore the thermal issues at an early stage of the design cycle and also during the physical design of MCMs. Proper heat management often becomes as vital as minimizing netlength (delay and area). *This is our main motivation in considering thermal constraints when we try to place the dies on an MCM substrate.* This approach is described in detail later in this chapter.

3.2.1 Approach to the Thermal Management problem

Presently, the CAD tools used for MCM layout (Mentor Graphics MCM Station and other EDA tools) operate in isolation with the placement being performed separately from the thermal optimization. The placement tool forces the dies to be close together while the thermal optimization tool often spreads them apart. Hence, it is desirable to consider both of these criteria simultaneously in order to obtain in less time the best solution which minimizes netlength yet still meets the thermal constraints.

The main goal of the work in this thesis was to integrate these two criteria

to provide a superior auto-placement tool. The interactive placement of dies on the substrate followed by a thermal analysis and rearrangement and followed by another placement and repeat thermal analysis as envisaged in the use of traditional MCM placement approaches, is not feasible when the number of dies to be placed on the substrate is large. However, considering multiple criteria simultaneously increases the computational burden and so there is a need to implement this optimization methodology in a parallel or distributed computing environment. This is described in detail in Chapter 5.

3.3 Problem Formulation for Placement

We redefine the MCM placement problem for our case as below:

Given a set of chips C and a set of sites S for these chips, find a mapping ϕ such that: C maps to S subject to timing, area and thermal constraints such that the number of layers needed for routing is minimized [19].

A set of slots or locations $\mathcal{L} = L_1, L_2, \dots, L_r$ are also given where $r \geq p$ (p is the number of dies). The placement problem is to assign each v_i to a unique location L_j so as to optimize some cost function. Each IC is assigned a slot L_j on the placement grid. The center of the chip is defined by the coordinates (x_j, y_j) which is the center of the slot L_j . One metric in the cost function is netlength. The wirelength metric is calculated based on the half-perimeter of the smallest

rectangle enclosing all points on a particular net. The calculation of the actual netlength is not possible until the routing phase is completed.

Let M be a grid array with r rows and s columns, and $r \cdot s \geq |V|$, of slots for chips. The objective is to assign the nodes of the graph G to the slots of M , known as a placement p , so as to minimize:

$$\sum_{i=1}^{|E|} c(e_i, p)$$

subject to the constraint:

$$\sum_{v_i \in V_N(i)} h(v_i) < H_{max}$$

where $c(e_i, p)$ is the cost (length) of edge e_i for placement p , $h(v)$ is the heat dissipation associated with chip v , and $V_N(i)$ is a set consisting of the the chips neighboring the chip v_i , and v_i itself.

Figure 3.1 shows the placement of ICs on the MCM substrate [19]. To achieve a uniform distribution of the heat over the MCM substrate, the average heat generated by all the ICs is first calculated. To determine that the placement like Figure 3.1 satisfies the heat constraint, the neighbors for each IC are determined and the average heat for the immediate neighbors for that particular placement is calculated. If this value is greater than the system heat figure then there is a heat penalty for that particular chip and placement. Otherwise, the heat penalty is zero. The heat penalty part in the cost function is minimized using the simulated annealing optimization approach. This method of calculating the thermal

constraint ensures that the components that dissipate large quantities of heat are placed far from one another.

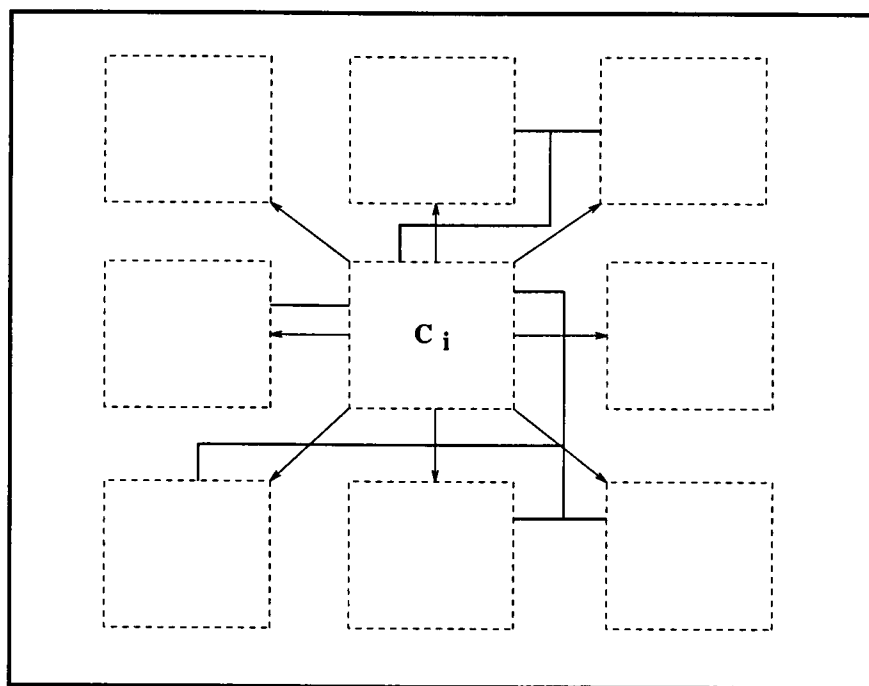


Figure 3.1: *Placement of Dies on the MCM Substrate [42]*

The MCM placement tool can either take the dissipated heat as the input or estimate the heat generated in the individual ICs in the MCM based on switching activity and the size of the dies. The temperature developed in each chip can be estimated as a function of the bonding technology, die characteristics and heat transfer methodologies available.

3.4 Placement as an NP-complete Problem

MCM placement belongs to the category of NP-complete or NP-hard problems [13]. These types of problems do not have a polynomial time algorithm. Physical design problems in VLSI and multichip modules are NP-complete problems. Researchers have classified all solvable problems into two classes : P and NP-complete. The P class of problems can be solved by deterministic turing machines in polynomial time. A deterministic turing machine is like a conventional computer. On the other hand NP-complete problems can be solved in polynomial time by non-deterministic turing machines. A non deterministic turing machine can be thought of as a parallel computer which can start up as many parallel processors as may be required by the number of different outcomes. Each of the new processors pursues the solution for one of the outcomes. However this type of model is non-realistic and non-feasible. The topic of NP-completeness is just dealt with here in brief and a good source for further reading is [13]. If every problem in class NP can be reduced to a problem P then problem P is in class NP. Reference [13] contains a list of some of the very famous NP-complete problems.

3.4.1 Algorithmic Approaches to NP-hard Problems

If a problem is proven to be NP-hard like the problems in VLSI design designers are left with the following choices to choose from [8]:

- Develop an exhaustive search approach where the whole design space is searched to find out the global solution. However this approach is not practically feasible for large design problems like a problem of placement of large number of dies on the MCM substrate.
- Develop algorithms based on heuristics. This type of approach gives us a non-optimal solution.
- Consider special cases of the problem and try to solve the problem.
- Use techniques for approximation.

3.5 Simulated Annealing Algorithm

Of late there has been a surge of interest in the use of a specialized class of random search algorithms called *simulated annealing* for optimization. Simulated annealing is a version of a thermodynamic statistical model that has been used to characterize the growth of crystals [12]. The solid state material is heated to a high temperature until it reaches an amorphous liquid state. Then it is cooled very slowly according to a temperature schedule. From the laws of thermodynamics we know that the thermal equilibrium at a temperature T is a probability

distribution in which a state with energy E is attained with the *Boltzmann* probability.

$$\frac{1}{Z(T)} \exp\left(-\frac{E}{K_b T}\right)$$

where $Z(T)$ is a normalization factor and K_b is the Boltzmann constant. The idea of simulated annealing is to transpose this model into the context of optimization problems. However this approach is completely based on heuristics and is not backed by a solid theoretical model. Simulated annealing works well for many types of combinatorial problems as long as sufficient computing resources are available.

In the problem formulation the energy is replaced with the cost function to be used for optimization and the states of matter are replaced by the legal states/instances of the problem or also the illegal states taken with a penalty of high costs. The temperature is evaluated with a control parameter that is also represented by T and is called the *temperature*. The general structure of the simulated annealing algorithm that we have used is given next. The simulated annealing algorithm is sometimes called a probabilistic hill climbing algorithm because not only does it allow decrease in values of the cost function during the optimization process but also allows increases in the cost function with a reduced probability. This way it helps to overcome local minima when we are actually looking for the global minimum.

Simulated Annealing Algorithm Used for Implementing MCM Placement

procedure Simulated Annealing

T = Tmax;

choose a random initial configuration;

calculate cost, C;

repeat

generate configuration;

repeat

perturb configuration;

$\Delta C = \text{new_cost} - \text{previous_cost}$

if $\Delta C \leq 0$ (downhill move)

then

accept that configuration $\text{prev_cost} = \text{new_cost}$;

else if

$\Delta C \leq 0$ (uphill move) and $\exp \Delta C/T \geq \text{random}[0, 1]$

then

accept the configuration $\text{prev_cost} = \text{new_cost}$;

until thermal equilibrium;

T := Update(T) (Update temperature according to temperature schedule);

frozen(T) (until frozen state, decides whether the final value has reached);

where,

Tmax : maximum temperature

T: represents temperature

C : represents the overall cost (thermal plus netlength)

3.6 Cost Function for Optimization

The objective of MCM placement is to place dies on the MCM substrate such that the delay for a signal in traveling from one point of the MCM substrate to another is reduced, the total area over which dies are placed on the substrate is reduced and a uniform thermal profile is maintained on the substrate. We use two different objectives in the cost function for the simulated annealing algorithm.

They are:

- The first objective is minimizing the total wire length used for interconnecting the various chips of the design, but no heat constraints are considered.
- The second objective is to minimize the total wire length but with a constraint on maintaining uniform thermal profile over the MCM (reducing the heat penalty).

3.6.1 Calculation of the Cost Function

The heat penalty part of the cost function is determined using the comparison of the average heat due to a particular set of neighbors and the system average for all the dies on the substrate. A placement that has less heat penalty than another but having the same wire length should yield a higher fitness value than the other. The final cost function reflects the total length of the interconnection wiring and also a number for how uniformly the heat is distributed over the MCM in terms of a penalty factor.

Calculation of the netlength of multi-terminal nets is a complicated procedure. We however use approximations here to calculate the netlength as have been shown in [6]. We calculate the length of a net by calculating the *half-perimeter* of the rectangular bounding box enclosing all the terminals that are connected by the net. This approach is shown in Figure 3.2. There are five terminals that the net connects and they are enclosed by the bounding box of dimensions x and y . The total length of the net is approximated as the half perimeter of the rectangle surrounding the pins connected by the net and is equal to $x+y$. This approach is also used by [19] and [5] for estimation of netlength for MCM placement.

3.7 Moves Used to Attain an Optimal Solution

A new configuration is obtained from three types of moves given next page.

- **Change Orientation:** The position of the die is changed by rotation of 90 degrees, 180 degrees and 270 degrees.
- **Exchange Positions of the chips on the substrate :** The position of two randomly selected dies is interchanged to see if there is any improvement in netlength.
- **Exchange position of a chip with empty slot:** A die is moved from a current position to an empty slot.

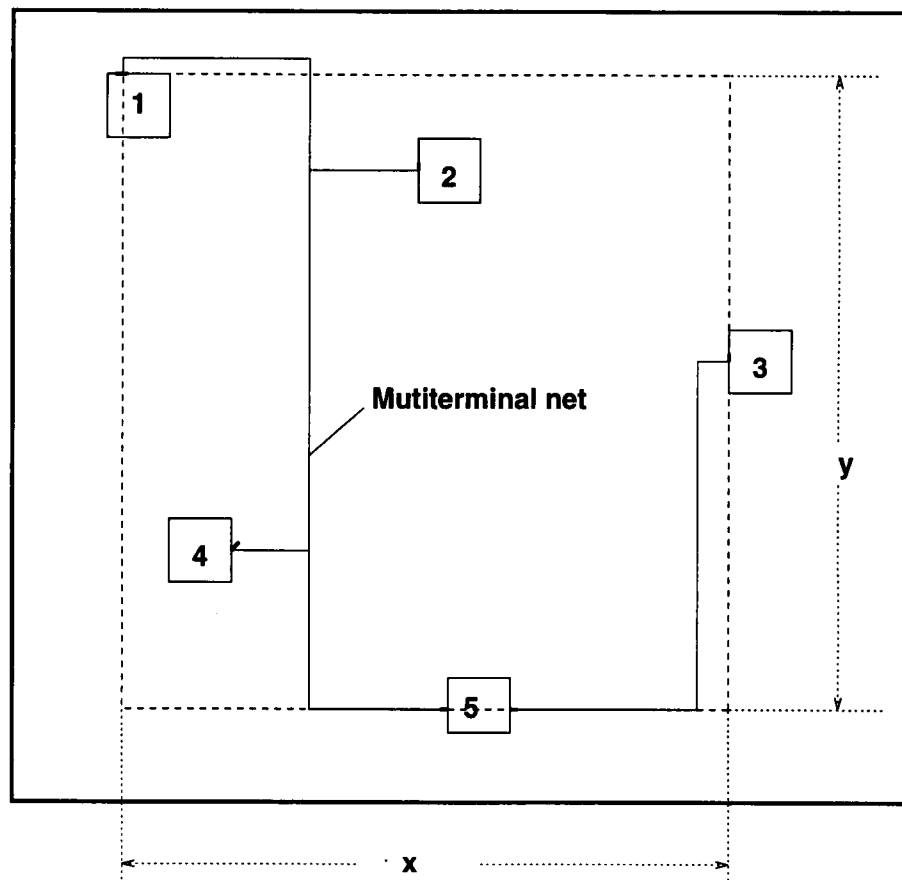


Figure 3.2: *Estimation of Netlength using Half Perimeter Method*

The use of moves or transformations is dependent on the temperature of operation and is an indication of how much progress has been made into the search based on a schedule.

3.8 Electrical/Thermal Placement Results

Table 3.1 gives the specifications of the designs that have been tested with this placement tool. Mcc2 is the Honeywell Supercomputer benchmark obtained from

Table 3.1: *Design Data for the Example Problems*

Design	No. of Dies	No. of Slots	No. of Nets
Mcc2	37	49	7118
Viper	10	12	1245
TLC	9	12	367
MM	7	9	802

MCNC, North Carolina. Viper Microprocessor and the traffic light controller (TLC) and Move Machine (MM) were obtained from the University of Cincinnati, Ohio. These input design netlists are in MCC format. The placements were done with and without the thermal consideration taken into account. A layout was generated for the initial randomized placement and the final placement of the ICs on the MCM substrate in each case. Since the two algorithms being compared are probabilistic algorithms, comparing the results obtained by one run for any

Table 3.2: *Results for Placement with the Thermal and Netlength Criteria*

Design	Wire_cost	Heat_cost	Total_cost	Time Taken (sec)
Mcc2	4.5991e+06	779.592	4.67761e+07	9939
	6.4726e+06	599.32	3.59598e+07	9993
	5.3212e+06	542.29	3.25379e+07	13888
Viper	1.7593e+07	33.333	1.99267e+07	56
	1.8878e+07	16.6667	2.00447e+07	55
	1.4664e+07	16.6667	1.58308e+07	55
TLC	1.4489e+06	0	1.44895e+06	43
	1.4003e+06	0	1.40037e+06	46
	1.4806e+06	0	1.48069e+06	53
MM	4.44038e+06	44.44	7.55149e+06	28
	4.6416e+06	0	4.6416e+06	27
	5.46643e+06	0	5.46643e+06	30

design does not make sense. Each algorithm was used several times for each design and the average values obtained by all the runs is shown in the tables that follow. For example Table 3.2 shows the netlengths and the heat penalties for the four test examples ie., mcc2, viper, TLC and MM for three best cases when both the netlength and thermal criteria was considered simultaneously. Similarly Table 3.3 shows the results when the netlength criteria was considered alone.

Table 3.3: *Results for Placement with only the Netlength Criteria*

Design	Total_cost (Wire_Cost)	Time Taken (sec)
Mcc2	1.68238e+06	12249
	3.68353e+06	13168
	4.01525e+06	12862
Viper	1.77756e+07	99
	1.75291e+07	95
	1.79181e+07	99
TLC	1.24453e+06	24
	1.46341e+06	26
	1.35423e+06	20
MM	4.22969e+06	27
	4.25209e+06	29
	4.53203e+06	30

Table 3.4 shows the comparison of netlengths and heat penalties for the placement with and without the thermal criteria considered simultaneously with the netlength minimization. This table indicates very important results from the point of view of justifying simultaneous consideration of netlength and thermal criteria for placement.

Table 3.4: *Comparison of results With and Without the Thermal Criteria (average netlength and heat penalty)*

Design	Without Thermal Constraint		With Thermal Constraint	
	Netlength(μ)	Heat Penalty	Netlength(μ)	Heat Penalty
Mcc2	2.9681e+08	1437.19	3.3051e+08	431.916
Viper	1.7542e+07	341.4	1.8867e+07	16.667
TLC	1.2445e+06	151.2	1.4489e+06	0
MM	4.25209e+06	247.222	4.6416e+06	0

The general equation for the total cost is,

$$Total_cost = Wire_cost * Wire_weight_factor + Heat_Cost * Heat_weight_factor$$

The heat_cost and the wire_cost are scaled using weights to meet the requirement that each of them are of the same order of magnitude. The wire_cost is just the total netlength in microns. The heat_cost is the heat penalty in watts. For the example of the Viper processor, heat_weight_factor is set to 70,000 and the corresponding wire_weight_factor to 1.0. These weighting factors can be varied to alter the relative importance of the thermal and wirelength constraints. The MAGIC layouts can be obtained for the best placement. The output is also available in the MCC format with the final position of the chips on the MCM substrate. MAGIC layouts with and without the thermal criteria taken into consideration are given in a following section.

3.9 Multi-Objective Optimization

By looking at the results in the last few pages we can see that a tradeoff analysis tool is needed for choosing the placement with a minimized wirelength and optimal heat penalty. A Multi Objective Design Advisor (MODA) [17] can be used to find out the *near optimal* placement configuration. The results from some of the test cases above were evaluated with MODA using the *tradeoff window*. For example for the Viper case we have 20 different placement configurations with different values of the total netlength and the heat penalty. The tradeoff window in Figure 3.3 shows the tradeoff results for 20 placement configurations. The selection of the optimal solution now depends on the design criteria. The *global optimum* or *near global optimum* point is the point closest to the origin, which represents zero heat penalty and zero wirelength cost. This point is marked as *global optimum* in Figure 3.3. This figure was generated using MODA [17]. Different strategies for removing heat are necessary for different points on the curve. For example a very hot MCM may require liquid coolant and a special heat sink, whereas a slightly cooler version might only require some type of fan or a heat sink. Heat sink design can be another intricate problem of its own. Cost and size considerations of different schemes might influence the decision as to which point is optimal or near optimal for given cost or size budget. A tool like MODA guides the designer through the process of selecting the best design from among many candidate solutions that are available. It is also necessary sometimes to look at a few good candidate solutions instead of just the best solution.

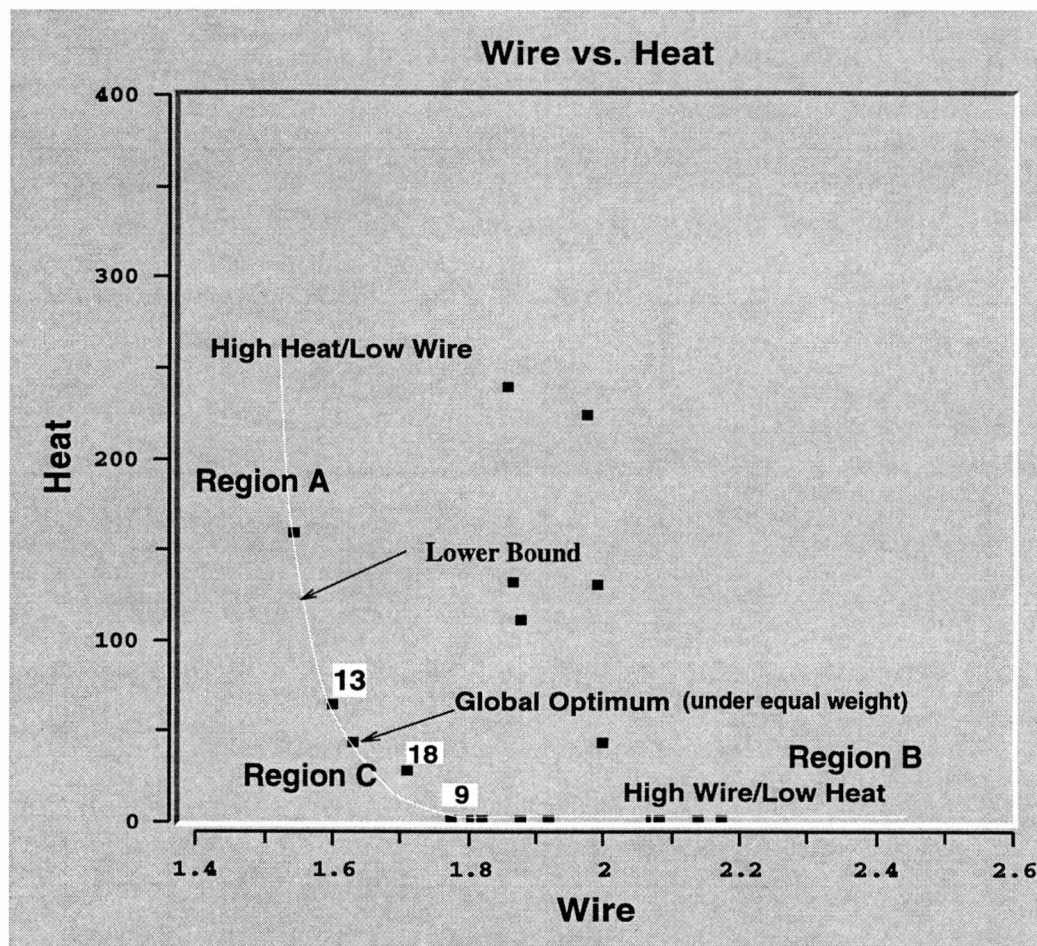


Figure 3.3: *Tradeoff Analysis Curve for the Viper Placement using MODA*

If the system speed is of vital then the version with the least total netlength may be the best choice despite the fact that the design will have a small area with the dies closely packed and one of the more elaborate cooling techniques may become necessary to remove a large amount of heat from such a small space. (Since the overall netlength affects both the delay as well as the area). From Figure 3.3 we can see that candidate number 13 is the near optimal solution. The next best solution is candidate number 18 or candidate number 9. It is slightly away from the *near global optimum point* and shows that there is a slightly higher weight for heat compared to netlength. Figure 3.4 shows the candidate solutions ranked according to their respective scores when the weight on the heat was half of what it was previously. From Figure 3.4 we can see that candidate 13 represents the best solution for the above conditions. We can also see that candidate 11 is a close second. This version shows that heat has been sacrificed a little to obtain smaller netlengths. If however we were to sacrifice netlength to obtain a more uniform heat/temperature distribution ie., a lower heat penalty we would choose candidate number 9 or candidate number 18.

3.10 Improvement in System Design Time

By considering multiple design criteria simultaneously while doing the placement of dies, we are able to reach the global optimum as defined in the Figure 3.3 faster. We can see how this happens by looking at Figure 3.5.

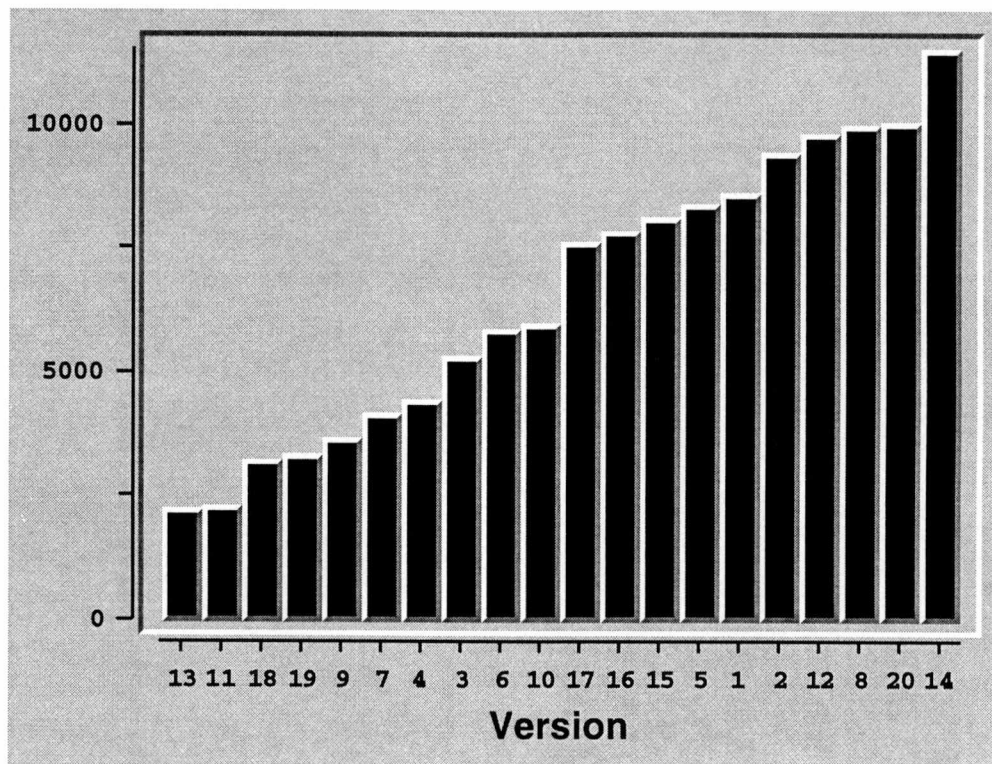


Figure 3.4: *Selection of the best candidate solution using MODA*

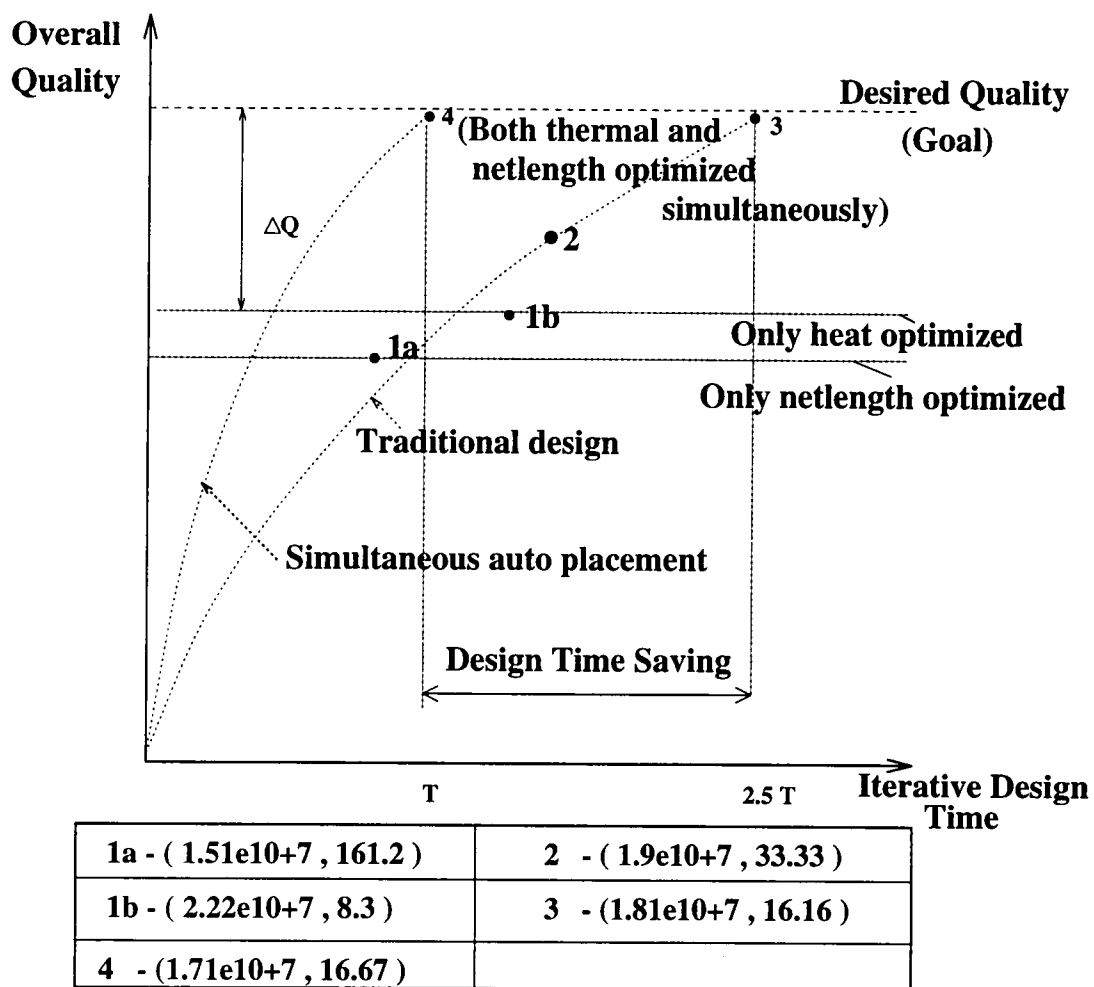


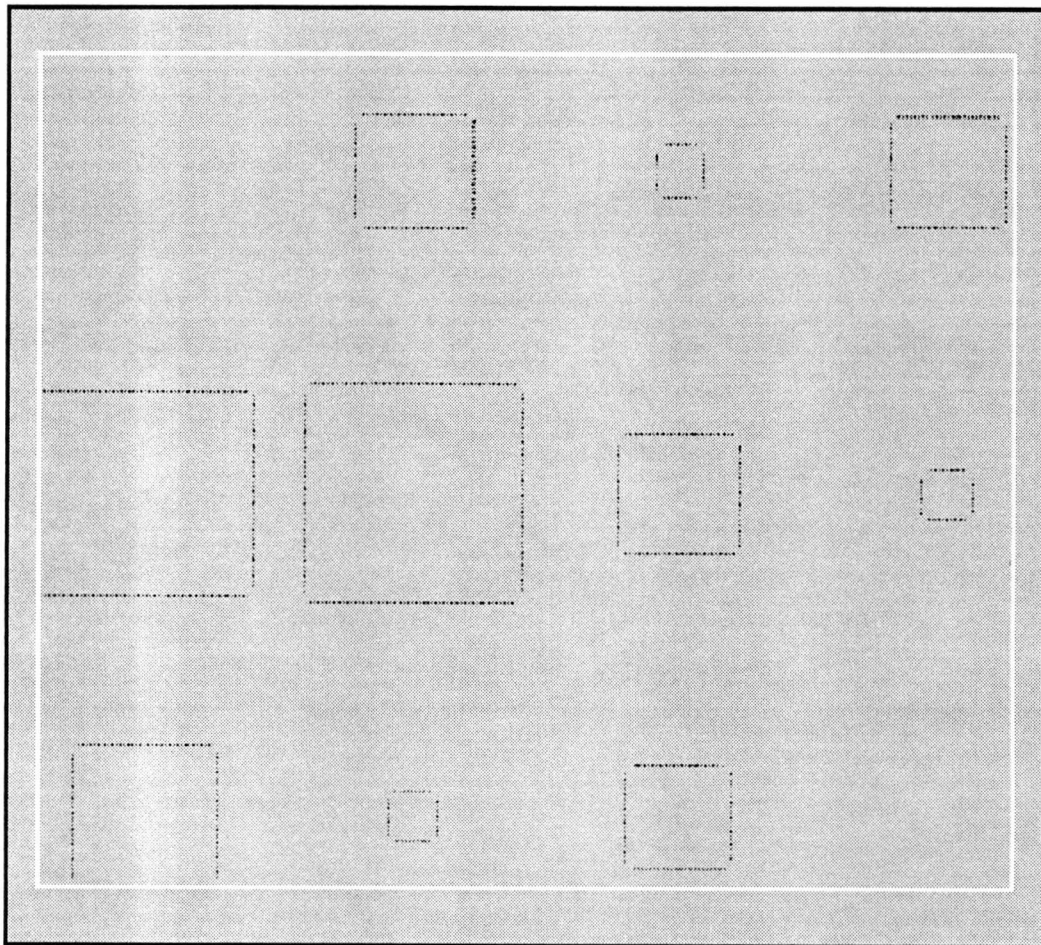
Figure 3.5: *Improvement in System Design Time*

Figure 3.5 shows a plot of the variation of overall quality with iterative design time. When only one objective is optimized, for example say netlength, we get a point in the design space in region A in Figure 3.3. This corresponds to *point 1a* in Figure 3.5. Suppose we optimize only for thermal penalty (heat penalty) we get a point in the design space in region C (refer to Figure 3.3). This point corresponds to *point 1b* in Figure 3.5. However the overall quality is below the desired quality goal. So we have to optimize further for netlength and maybe reach the *point 2*. However since we are using mutually opposing criteria separately it will take us a long time to reach the desired quality level (*point 3*). However by considering multiple criteria simultaneously we can get a point in the design space close to the global optimum point as given in Figure 3.3. This point is shown as *point 4* in Figure 3.5. We can see by considering multiple criteria simultaneously we can reach the desired quality level much faster. These values of iterative design time will vary depending on the type, size and number of nets in the design.

3.11 Analysis of the Results

The wiring (netlength) and the thermal costs for the placements of the three example designs are given with the thermal consideration taken into account. As explained earlier the thermal value for each component is calculated for a placement and if this value is greater than the system average value then a penalty is given. The results in Table 3.2 shows that heat_cost contributes to the total cost for each placement since it has been scaled to be the same order of magnitude

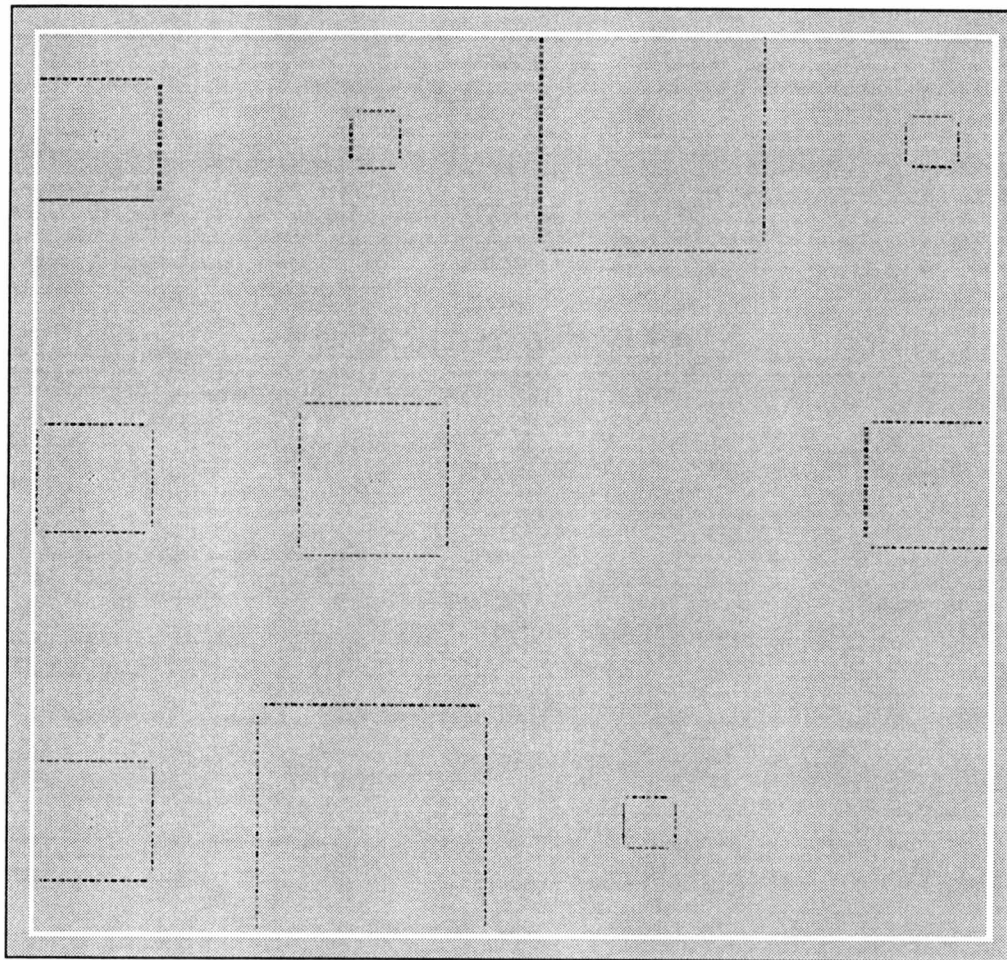
as the `wire_cost`. As mentioned in an earlier section we can see that Table 3.3 shows the placement costs without the thermal criteria considered. Here the total cost is just the `wire_cost` or `netlength` cost. If the thermal constraint is also considered the placement tool tries to place the chips/dies farther apart than it would have normally done without the thermal constraint resulting in a larger `netlength` or `wire_cost`. Looking at Table 3.4 we can see the difference in the `wire_cost` when considering the thermal criterion or neglecting it. We can see that the `wire_cost` is larger in all the cases when the thermal effect is considered. Also we see that overall heat penalty differs in both the cases. The numbers in the column *overheating* do not have any significance by themselves. We can see that there is a substantial reduction in overheating by considering the thermal effect. Figure 3.6 shows the placement of the dies on the substrate without the thermal effect taken into consideration. Figure 3.7 shows the placement for the *Viper* with the thermal effect taken into consideration. Figure 3.8 shows the placement layout for the *mcc2* example with only the `netlength` minimization taken into consideration whereas Figure 3.9 shows the *mcc2* placement layout with both the thermal and the `netlength` minimization taken into consideration. By looking at these two figures we can see clearly that the two large dies which have a the highest power dissipation are placed further apart. However, this is not the case with placement without the thermal effect taken into consideration.



Netlength = 1.75e+07 microns

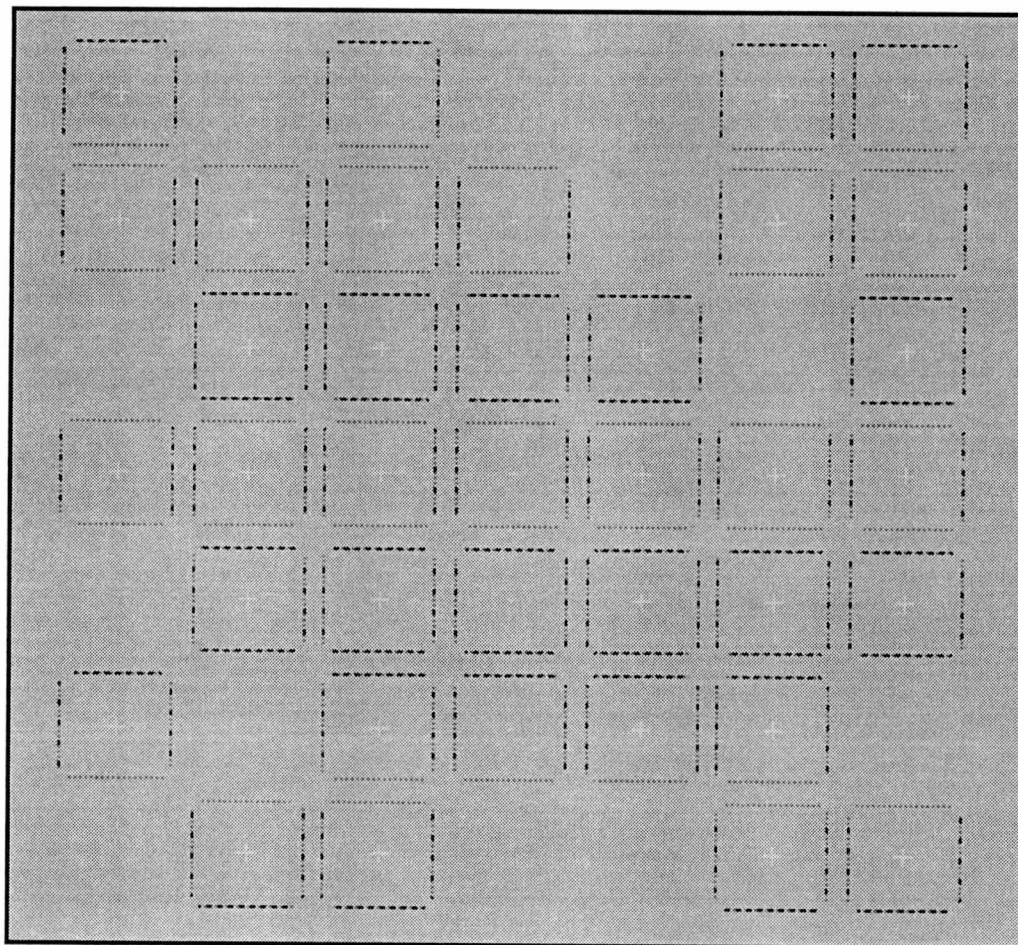
Heat Penalty = 341.2

Figure 3.6: *MAGIC Layout for Viper on the MCM Substrate without Thermal Considerations*



Netlength = 1.88e+07 microns
Heat Penalty = 16.167

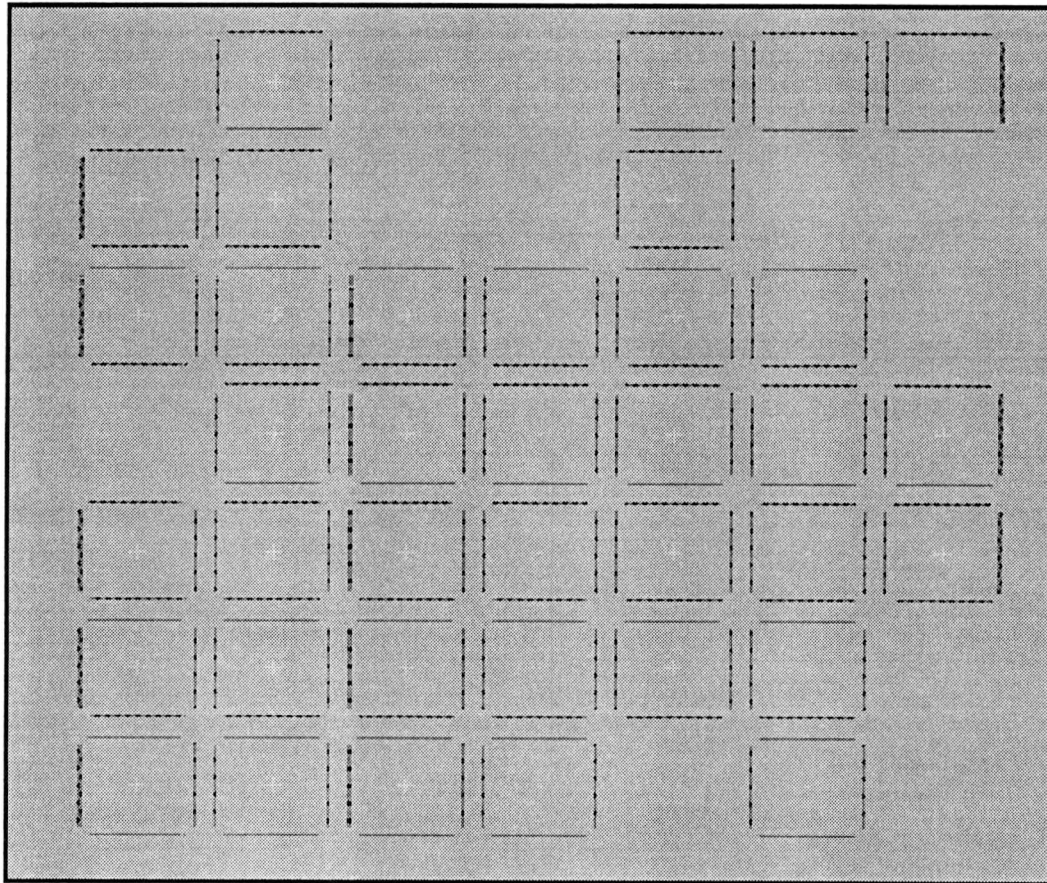
Figure 3.7: *MAGIC Layout for Viper on the MCM Substrate with Thermal Considerations*



Netlength = 2.97e+08 microns

Heat Penalty = 1437.19

Figure 3.8: *MAGIC Layout for mcc2 on the MCM Substrate without Thermal Considerations*



Netlength = 3.30e+08 microns

Heat Penalty = 431.91

Figure 3.9: *MAGIC Layout for mcc2 on the MCM Substrate with Thermal Considerations*

3.12 Conclusions

We can see from the results in this chapter that by sacrificing a little on the netlength we can obtain a much more uniform heat distribution pattern on the substrate. This uniform heat distribution pattern is a necessity in multichip modules. We have also shown that the design time is reduced considerably by considering the netlength minimization and the thermal criteria simultaneously in the placement tool. This difference in design time becomes more significant in the case of large designs. The approach here is that for a two dimensional optimization problem. However we may add other objectives to the placement objective and have a true multiobjective optimization problem. Timing delay driven placement for placement of dies on the MCM substrate along with netlength minimization and thermal criteria will lead to further improvements. As the cost function becomes more complicated the need for running the optimization in a parallel distributed environment becomes more relevant and essential especially for large designs. A more accurate approach to the placement problem will involve consideration of the different bonding technologies, heat sinks etc., in more detail when calculating the internal and the external thermal resistance for each of the dies. This information can be used to calculate the temperature developed at the site of each die due to its internal heat generated and also due to the effect of heat transfer from the other dies through the substrate.

CHAPTER 4

COMPLETE MCM PHYSICAL DESIGN USING MENTOR GRAPHICS MCM STATION

4.1 Introduction

In this chapter complete MCM design using Mentor Graphics MCM Station has been described. MCM design using Mentor Graphics Corporation, MCM Station V8 has been used for some years now for the design of multichip modules. The MMS/MGC technology kit has been available since 1994 as part of a design support system for developing multichip modules. The placement information obtained from results in the last chapter has been imported into MCM Station to complete the routing and hence complete the physical design cycle. the completion of the physical design cycle is the main goal in this chapter.

The MCM Station from Mentor Graphics uses several toolboxes like *Layout Tool*, *Librarian* and *Autotherm* to go through all the steps in the MCM physical design process. The MCM Station tools take as an input the partitioned mask layout files for each of the dies that are to be placed on the MCM substrate. The cell geometry information is written out from the MAGIC layout tool in the form of GDSII data and read into Mentor Graphics MCM Station. The die maps and

corresponding netlists can also be generated and imported into the MCM Station. This saves a lot of time which otherwise needs to be spent in creating cell and substrate geometries. The next stage in the physical design process after reading in the cell geometries is *placement* of dies on the MCM substrate. The Mentor MCM Station tool *Layout* is used for this purpose. During the placement step the placement process is completed either automatically or interactively. This is done interactively in most cases. This makes the optimization of the netlength quite difficult especially if the number of dies on the MCM substrate is large. Also the thermal distribution constraint is not taken into consideration during placement of the dies. After the placement is completed the die geometries are exported to another MCM Station tool *Autotherm*. This tool does the FEM (Finite Element Method) based thermal analysis to find the die junction temperatures and also the thermal flow patterns on the substrate. After the placement satisfies the thermal constraints the *Auto Router* tool is invoked to achieve the final routing. These steps complete the physical design process. The placement and routing steps attempt to minimize the total area, signal delays and interconnect wirelengths. Figure 4.1 shows the design flow for MCM physical design using MCM Station. However many variations of this design flow have been adopted by designers. Most of the MCM designers of today acquire bare dies from the chip supply vendors and use them to design an MCM. After the placement is complete a thermal analysis is executed to check for uniform thermal distribution on the MCM substrate. After this, the design is routed using the *Autorouter*.

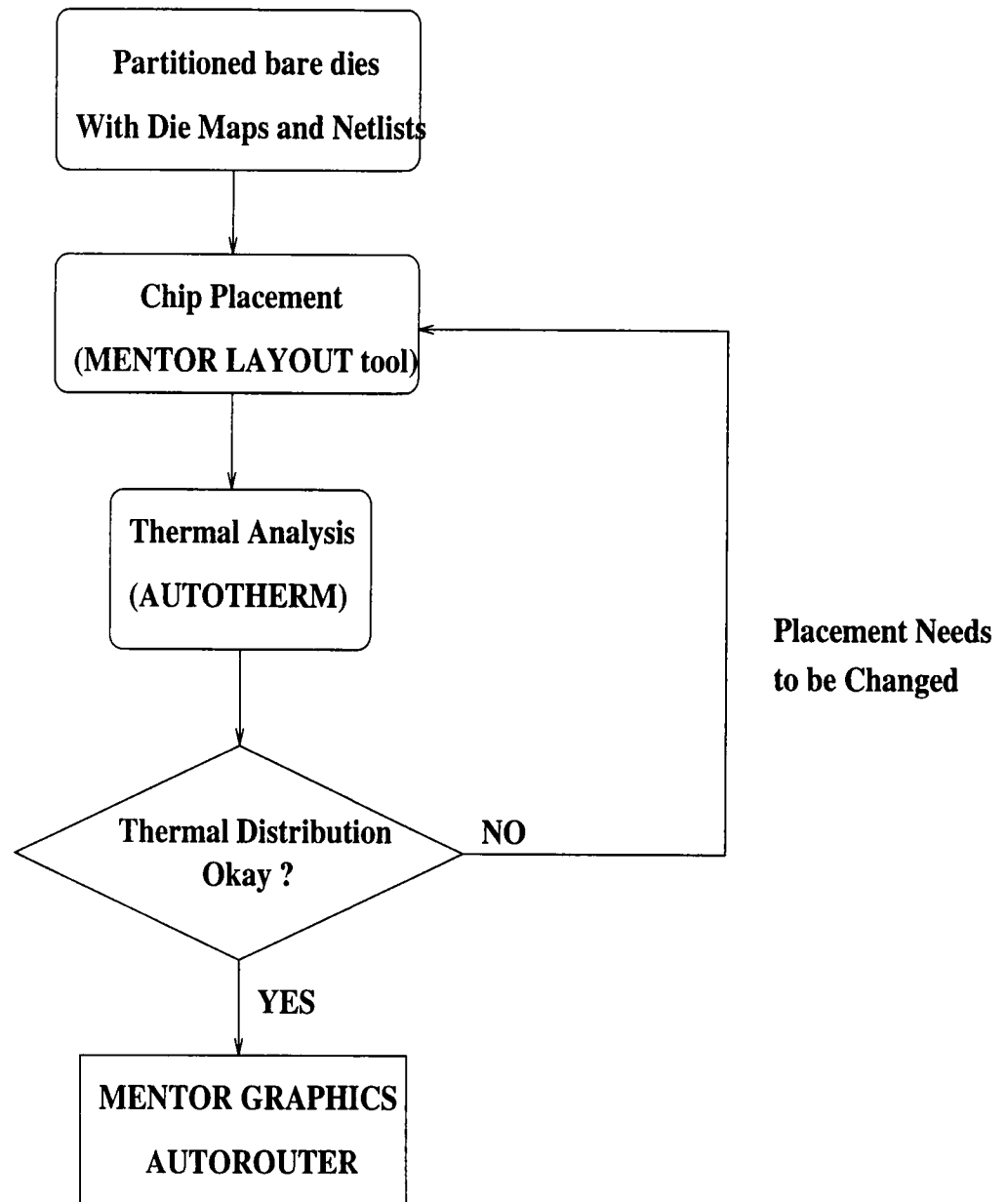


Figure 4.1: *MCM Physical Design - Flow using Mentor Graphics MCM Station*

4.2 The Design Objects Available in Mentor Graphics MCM Station

Following is a list of design objects that will need to be created before running any of the tools:

- *Layers design object*

The layers design object lists all the available layers and also lists some of their properties. There is a default layers object that can be used, but some of the geometries in the geometry design object use additional layers that need to be defined.

- *Tech design object*

This design object contains all the information such as design rules.

- *Geoms design object*

This design object is not necessary to start off with but there are several geometries already designed, such as vias, which can be copied from one design to another, rather than recreated again from scratch using the Mentor tool *Librarian*.

- *Nets design object*

This design object gives the connectivity of all the components in the MCM and is used by the layout tool to do the routing.

- *Comps design object*

This design object is a list of all the components in the design (including

I/O connectors) and, eventually, their locations on the board. The ascii version of this design object should be created by the partition engine.

4.3 LAYOUT Tool in MCM Station

4.3.1 MCM Placement

Placement of dies on the MCM substrate is an important part of the MCM physical design process as has been mentioned in the earlier chapter. It is necessary to have a good placement to ensure that the timing delay, area and thermal constraints are satisfied and the overall netlength (delay and area) is minimized. Since the density of designs in MCMs is very high the heat generated due to the close proximity of dies on the substrate needs to be distributed uniformly to avoid thermal hot spots and an eventual breakdown of the substrate. The two types of placement options available in MCM Station are listed below. The *Layout* has provision for *interactive* as well as *automatic* placement.

When the board is being designed the following placement rules have to be adhered to [3]:

- Placement outline
- Placement keepouts
- Placement regions
- Placement rules

- Placement clearance
- Valid component

Automatic Placement

The *Layout* tool has the provision for an automatic connectivity based placement or constructive placement as defined in Chapter 2. Automatic placement is used to improve the quality of the placement occasionally but is used for the initial placement most of the time. The auto placement function places components to reduce the connection netlength.

4.3.2 Interactive Placement

The *Layout* tool allows placement of dies on both sides of the board or the substrate. However we have done placement only on one side of the board. The interactive placement should generally be used after the placement has been done automatically for the first time to give a reference to improve upon. The *Layout* tool allows component dragging for quick interchange of positions and identification of optimum positions through the usage of forced vectors and guides. At any point of time the placement can be evaluated by using the routing predictor or the histogram display. For example Figure 4.2 shows the rats nest display showing the connectivities of the chips on the substrate. If the placement has already been determined by *Autotherm* then the user will have to be careful in placing the components to match the thermal results given by *Autotherm*.

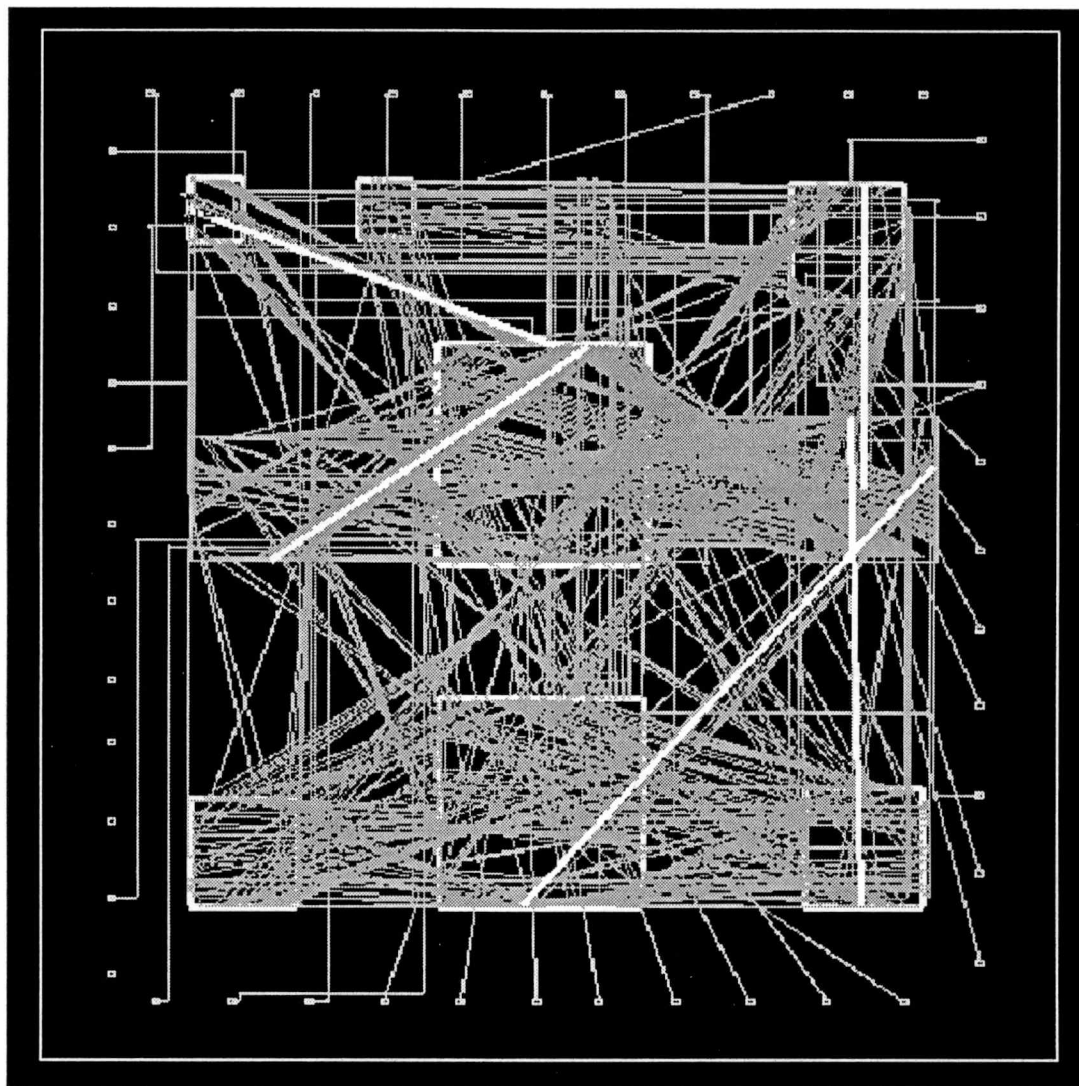


Figure 4.2: *Rats Nest Display for the Viper Placement*

4.3.3 Thermal Analysis After Placement with AutoTherm

After the components have been placed on the board or substrate it is necessary to run a thermal analysis for the placement to determine the temperature distribution on the substrate. The MCM Station tool *Autotherm* can be used for the thermal analysis. After the placement process has been completed the ASCII geometries are exported to the *Autotherm* tool. The power dissipation as well as the internal resistances of the dies to the substrate are to be given as input to the *Autotherm*. The internal resistance to the substrate (*therm_r*) depends on the power dissipation of the die, the type of bonding technology chosen, the type of substrate used (material) and some other factors. The power dissipation (*pow_typ*) is calculated based on the current, voltage and the switching activity. This is shown below. The thermal model for the die and the substrate is shown in Figure 4.3. Some of the definitions are given below.

$$Power_typ = V_{cc} * I_{cc} + P_O * V_{ol} * I_{ol} \quad (4.1)$$

Where,

V_{cc} is the power voltage

I_{cc} is the typical current drawn by the chip

P_O is the number of output pins in the component

V_{ol} is the typical output low voltage

I_{ol} is the typical output low current

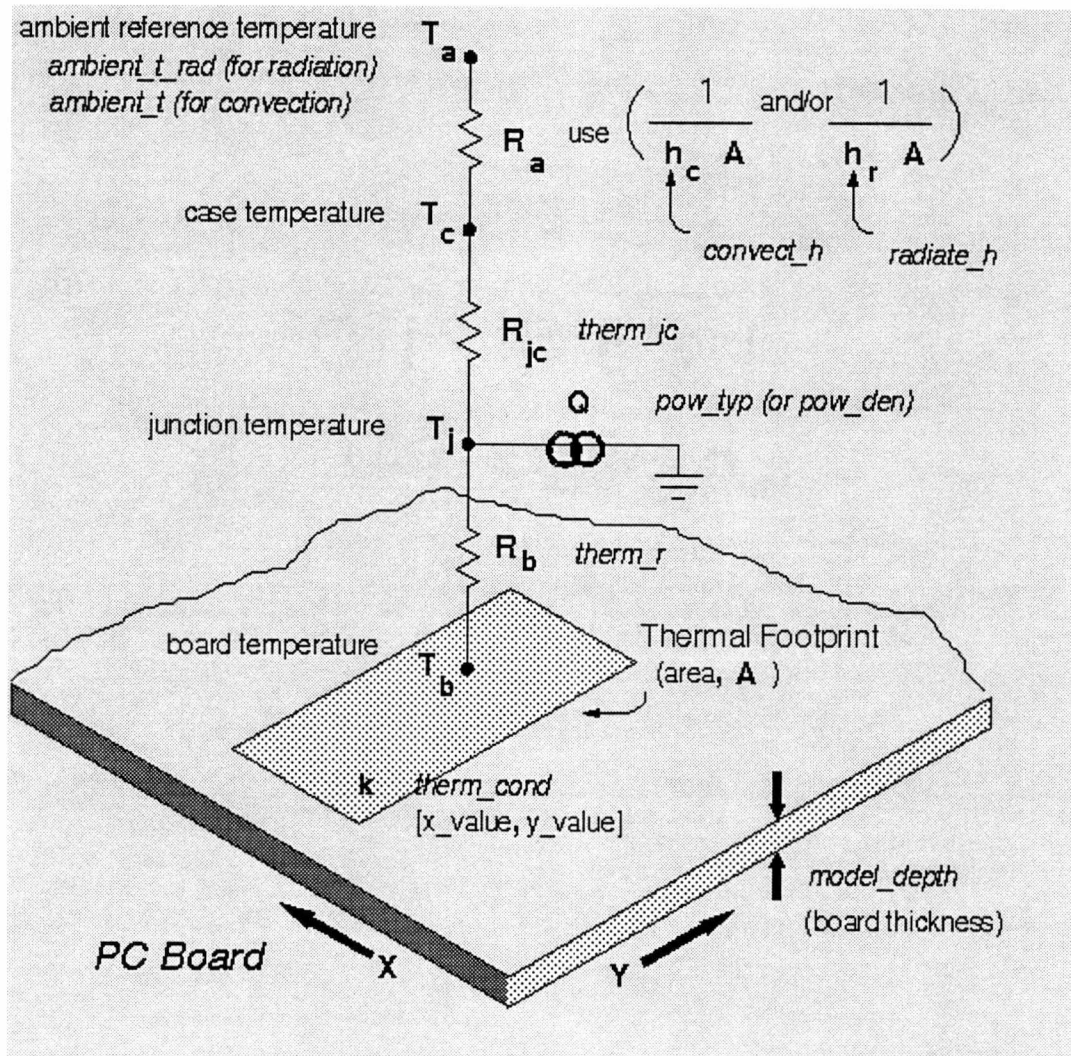


Figure 4.3: Thermal Model used in AUTOTHERM

After all the inputs are provided to the *Autotherm* tool and the types of heat flow, ambient temperature etc., are chosen the *Autotherm* solver is executed to generate the thermal flow patterns and the junction temperatures for the dies on the substrate. Figure 4.4 shows the junction temperatures for the dies. If the junction temperatures are more than what is desired the dies can be moved around on the substrate until the desired junction temperatures are achieved. However this changes the initial placement as determined using the interactive placement tool. Hence the netlength changes. It is difficult to obtain a true tradeoff between the thermal and the netlength minimization criteria using this method of doing the placement and the thermal analysis separately one at a time. Several iterations may be necessary before we get a placement that optimizes both the netlength and uniform temperature distribution objectives.

4.3.4 Auto Routing using the Layout Tool

Routing is the final step of the MCM physical design process. The router available in Mentor MCM Station is a grid based sequential maze router that can take very long run times depending on the grid size used for routing. The program flow for the autorouter is given in Figure 4.5, [3]. This router takes a very long time to complete the routing even for reasonably small designs. For example for the 10 chip Viper microprocessor that has around 1200 nets it takes about 10 hours to route on a Sparc Station 5. However there are shaped based gridless autorouters that are now available as third party tools integrated with MCM Station.

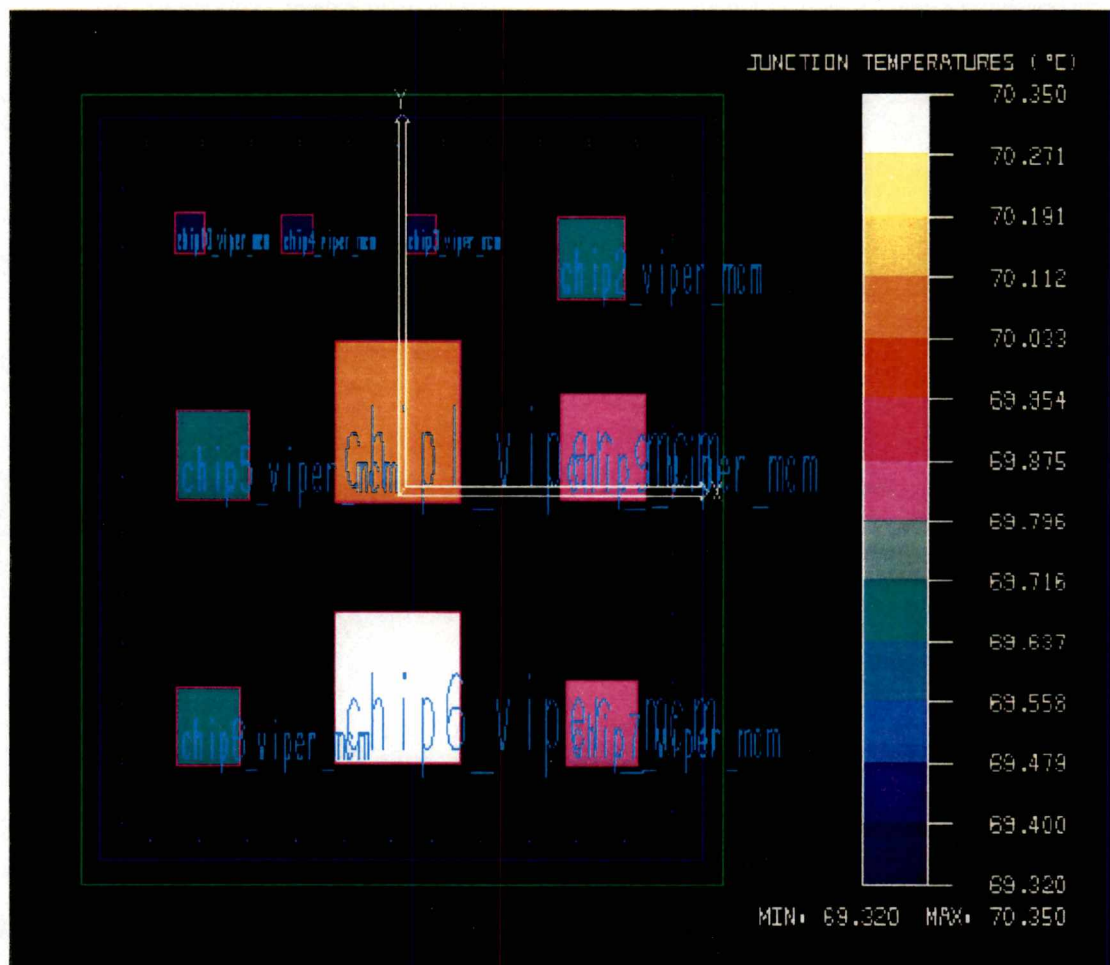


Figure 4.4: *Junction temperatures for Viper design*

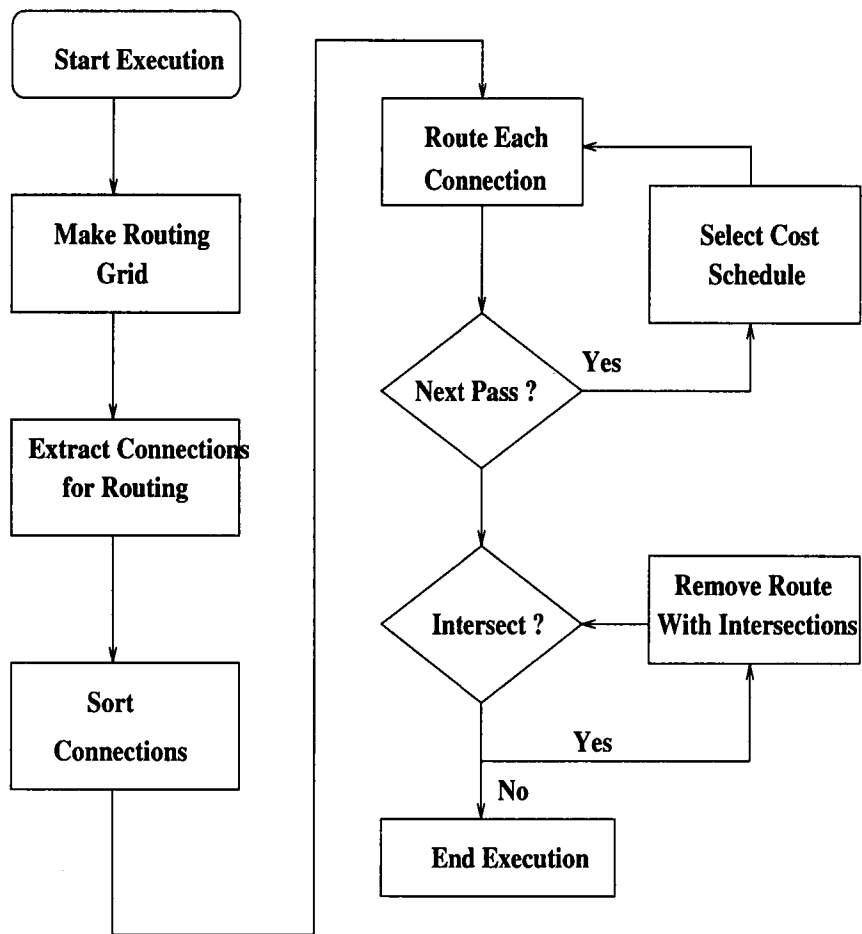


Figure 4.5: *LAYOUT, Autorouter Program Flow*

The grid based *autorouters* have the following features:

- Iterative - The number of passes that the tool will go through depends on the number of passes being set. This can be set up based on the requirements of the design. Each pass uses a different cost schedule to determine if a given routing is acceptable or not. It is recommended to use at least 4 - 5 passes to begin with. It has been mentioned that 25 passes will be sufficient to route any type of MCM design. The manufacturing router checks the errors and will rip up any bad route. One pass of this router should be sufficient.
- Maze Runners - Each connection is routed from source to target. The router moves along the grid and decides at each point which direction to move next. It uses a cost driven iterative algorithm with rip-up and retry capabilities. When diagonals are allowed the number of possible moves from a point on the grid is eight. When deciding on a move the router waits to see what the costs involved with each particular movement are.
- Rip up and retry - As it passes through the candidate solutions it can rip up and retry each connection with a new cost schedule. If it achieves a lower total cost it accepts that rerouting and discards the old one. If it does not succeed in getting a lower cost it reroutes the old connection.

Figure 4.6 shows the Viper design routed using *Autorouter*.

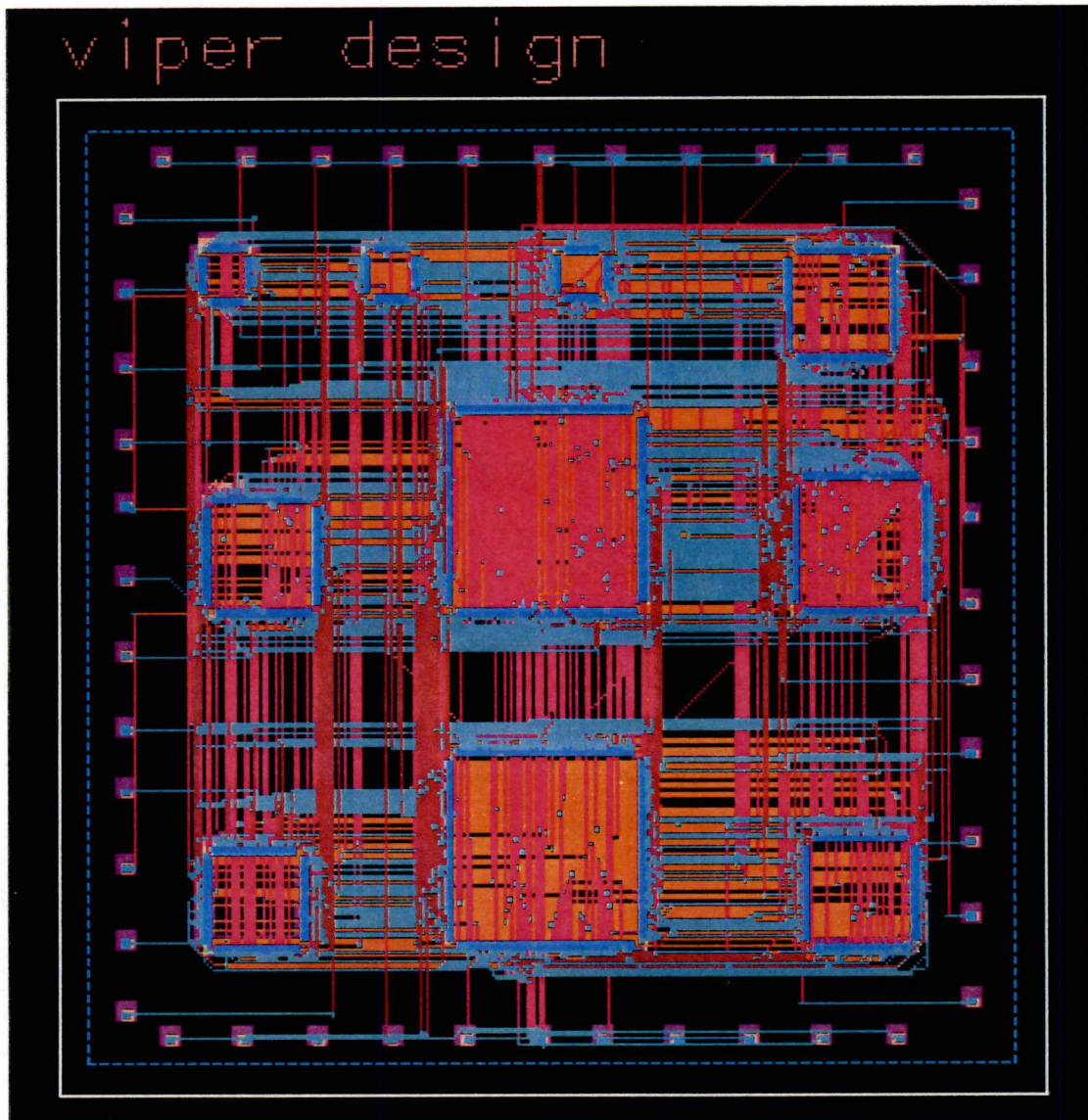


Figure 4.6: *Routed Viper Design*

4.4 MCM Physical Design Using Design Kits

Of late many design kits are available for the design of multichip modules. Some of the notable ones are AT&T (MCM-D), nCHIP (MCM-D), Hughes Aircraft (MCM-C,D), MMS (MCM-C,D), TI(MCM-D, HDI) and SOREP (MCM-C). Design kits have a lot of advantages for the substrate designer because of the fact that the design rules are set in the technology files, electrical parameters set in the technology file, physical layers are set in the technology file, etc. Sometimes ERC/DRC rule checkers are also provided. The MMS design kit helps us to generate templates for the substrate and dies instead of creating geometries using the *Librarian* tool from Mentor Graphics. Mentor Graphics MCM Station supports all the above tool kits.

4.5 Conclusions

From the approach to physical design of MCMs using MCM Station we can see that the placement step includes the placement of dies on the MCM substrate considering some design criteria like the netlength minimization followed by a thermal analysis and check of the placement. If a design violates the thermal constraints then the placement has to be changed and the thermal analysis redone to check if the thermal constraints are violated or not.

This *iterative* approach to design is not very desirable especially for a large design. What is necessary is an automatic way of combining the thermal and

the overall netlength minimization criteria to obtain a good placement. Thus the design cycle time can be reduced considerably and both thermal and netlength criteria can be considered depending on what design specifications are held as more important for a particular design. This approach was described in the previous chapter.

CHAPTER 5

PARALLEL IMPLEMENTATION OF THE SIMULATED ANNEALING BASED PLACEMENT USING PVM

5.1 Introduction to Parallel Virtual Machine (PVM) and XPVM

A Parallel Virtual Machine (PVM) is a software system that enables a collection of heterogeneous computers to be used as a flexible and easy to use concurrent machine [11]. The individual computers connected as a parallel virtual machine may be shared or local memory multiprocessors, vector supercomputers, specialized graphics machines or scalar workstations connected using Ethernet or FDDI. PVM executables run on different machines that are part of a user-configurable pool and present a unified general concurrent virtual machine. Programs written in C and Fortran can be used in a PVM environment by giving function calls to PVM library that contains routines such as process initiation, message passing and process termination. A PVM environment is ideally suited for concurrent applications with many interrelated parts. XPVM was developed as part of the joint PVM effort at Oak Ridge National Laboratory, University of Tennessee and Emory University. XPVM is written in C with the X-Window system,

using the TCL / TK toolkit. It provides a GUI for the PVM environment. It includes PVM console commands like tasks - spawn, kill, signal and signal tasks, reset, quits halt etc.

5.2 Simulated Annealing Algorithm in a PVM Environment

5.2.1 Advantages of using a PVM with SA Algorithm

There is always a tradeoff between the optimality of a final solution and the runtime requirements when one uses a heuristic or a probabilistic algorithm. Generally a heuristic algorithm with a larger time complexity generates a better solution for an optimization problem. The use of a PVM helps in the following ways:

- If we use the same runtime then a better quality optimization solution can be generated by using a PVM since one can use smaller steps and more complicated heuristics, compared to executing the optimization routines in a single machine environment.
- Fixing the step size or the heuristics used, an optimal solution can be generated much faster by using a PVM in comparison to a single machine environment. Problems of larger size can be solved much faster in a PVM environment.

5.2.2 Implementation of a PVM

The implementation of the simulated annealing algorithm for the MCM placement can be achieved easily using a distributed processing environment like PVM where different workstations are connected in a network. In [5] the author has attempted a distributed placement of dies in multichip modules. The MCM placement algorithm is iterative in nature and is computationally intensive for placement problems which have a large number of dies and nets (for example the *mcc2* design that has 37 dies and 7118 nets). In optimization procedures like MCM placement using a simulated annealing algorithm most of the time is spent in cost function calculation. The computational burden increases with the complexity of the cost function considered for optimization. This makes this problem an ideal problem to be implemented using a parallel virtual machine (PVM). This type of problem can be implemented using a scheme shown in Figure 5.1.

The simulated annealing based electrical/thermal placement tool was implemented in a parallel environment using PVM and the XPVM interface. The scheme operates with one *master* task controlling the actions of *n slave* tasks by passing messages and data between the master and slave processes. Thus for this application the optimization is carried out using a *master* processor and the cost function calculation is carried out using *slave* processes. The slave processors are normally idling and can be activated only on request from the master processor. The master processor synchronizes the slave processors using a single multicast made by the master processor.

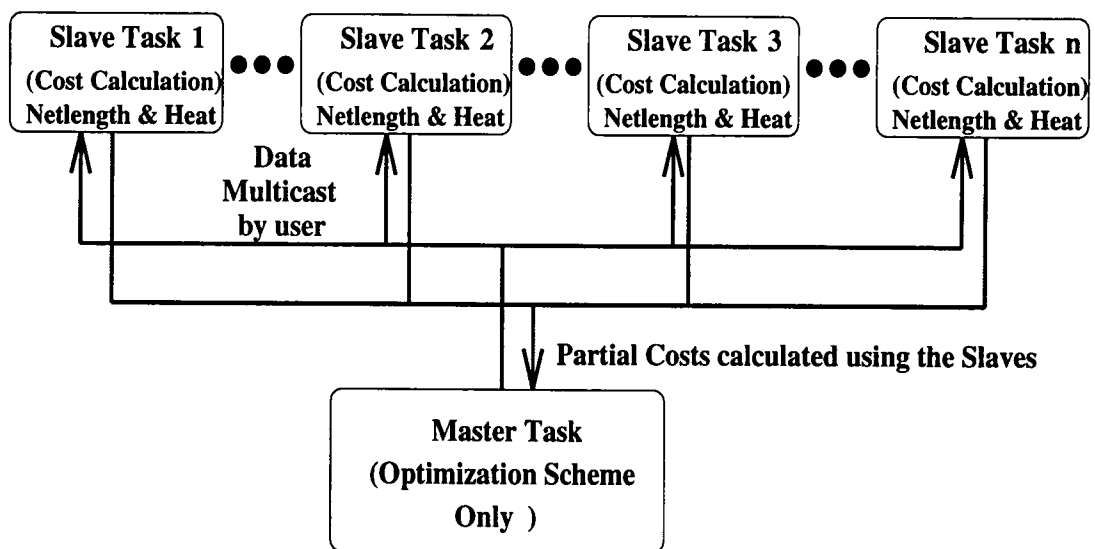


Figure 5.1: *PVM Implementation Scheme*

5.2.3 Parallel Simulated Annealing Algorithm for MCM Placement

We have implemented the simulated annealing based PVM implementation of the MCM placement optimization using a cluster of Sun workstations which include Sparc Stations - 10, 5 and 2. Some of the machines have 64Mb or 128 Mb memory while others have a 32 Mb memory. The PVM system implementation is divided into two parts. The *pvm* is the PVM daemon and runs on all the machines that constitute the Parallel Virtual Machine. The PVM can be started by executing the daemon on one of the machines in the network. This is accomplished by using the *pvm* executable in the `/pvm3/bin/SUN4` directory in the home directory where PVM is installed. Once the daemon is started on one machine other machines can be added to the PVM. Alternately the PVM daemon can be started on all the machines by using adding the machines in the *hosts* file. The second part of PVM is the PVM library routines to start up tasks, communicate between tasks, synchronize tasks etc.

The scheme operates with one *master* task controlling the actions of *n slave* tasks by passing messages and data between the master and slave processes. Thus for this application the optimization is carried out using a *master* processor and the cost function calculation is carried out using *slave* processes. The slave processors are normally idling and can be activated only on request from the master processor.

5.2.4 Implementation of the SA Algorithm

The cost function calculation for the simulated annealing algorithm involves calculation of the netlength and the thermal penalty. As stated earlier in the implementation of a SA algorithm 80 % of the time is spent in cost function calculation. Here the cost of placement affects the quality of placement. The calculation of the netlength itself is quite time consuming because for each change of position of the dies or the rotation of the dies there is an associated change in netlength for that die. So every time there is a perturbation in the simulated annealing algorithm there is a cost function calculation involved. PVM implementation attempts a concurrent calculation of the costs of placement. The *master* does the optimization and also spawns the other slave tasks that calculate the cost function in the other machines. Each of the slave tasks do a part of the cost function calculation for the optimization which runs in the master mode.

5.2.5 PVM Call Constructs in the Simulated Annealing Algorithm

The following PVM constructs have been used in the *Master* mode implementation of the simulated annealing algorithm:

- `pvm_spawn("slave_prog", arg_array, int flag, machine name, number of slaves, slavetids)`. This routine starts up an executable called "slave_prog" on the virtual machine. The *flag* option sets up the name of the machine where the PVM chooses to spawn the processes. We have used the *Pvm-TaskDefault* option to let the PVM decide where to spawn the processes.

pvm_spawn runs on the master machine in the *master* program. The *master* program runs the following routines.

- *pvm_initsend(mode)*, *pvm_pklong(data,size)*, *pvm_mcast(slavetids, number_of_slaves)*, *pvm_rcv()*, *pvm_unklong()* Sending a message in PVM is composed of three actions. First a send buffer must be initialized by a call to *pvm_initsend()*. Secondly the message must be packed into the buffer by using the any of the *pvm_pk** routines like *pvm_pklong*. Finally the message is sent using the *pvm_send()* routine or multicast using the *pvm_mcast()* routine. We have used these three routines in the *master* program. The data (which involves the partially calculated cost function values) is received by the *master* program using the *pvm_rcv()* routine. The received data is unpacked using the *pvm_upk** routines.

The slaves spawned by the master calculate the netlength and the heat penalty whenever the *master* multicasts the slaves. Upon receiving the data from the slaves the *master* calculates the total cost by adding the slave costs from all the slaves. The *slave* routines are :

- *pvm_parent()*, *pvm_rcv()*, *pvm_send()*, *pvm_upklong()*, *pvm_pklong()*

This process is just the opposite of what happens in the *master* program. The *slave* program gets the chip positions from the *master* program and then calculates the netlengths and heat penalty and then passes these values onto the *master*. *pvm_pklong* and *pvm_upklong* packs and unpacks the data.

5.3 Performance Comparison when using a PVM

Suppose we have $(n+1)$ workstations out of which one runs in the *master* mode and n of them run in the slave mode. If there are m nets then each slave task and the master task calculates the cost of $m/(n+1)$ nets. This shows that the computational burden in each of the workstations is reduced drastically. This savings in time is significant when the number of nets in the design is large. The PVM environment was simulated using an XPVM graphical user interface. Figure 5.2 shows the XPVM setup with the different menu items highlighted. Figure 5.3 shows the PVM spawn process. The different workstations shown in the network window form the PVM. The different machines which are active at any particular time is shown shaded. This lower window also shows the events and the machines involved at any time. Let us list some of the quantities we use to characterize the PVM environment as defined below:

$$\text{Speedup} : S = T_1/T_p$$

$$\text{Efficiency} : E = S/p$$

where,

T_1 : execution time with a single Sparc station

T_p : execution time with "p" Sparc stations

p : number of processors

Figure 5.4 and Figure 5.5 show the speedups obtained by using a distributed computing environment for the mcc2 and Viper examples respectively.

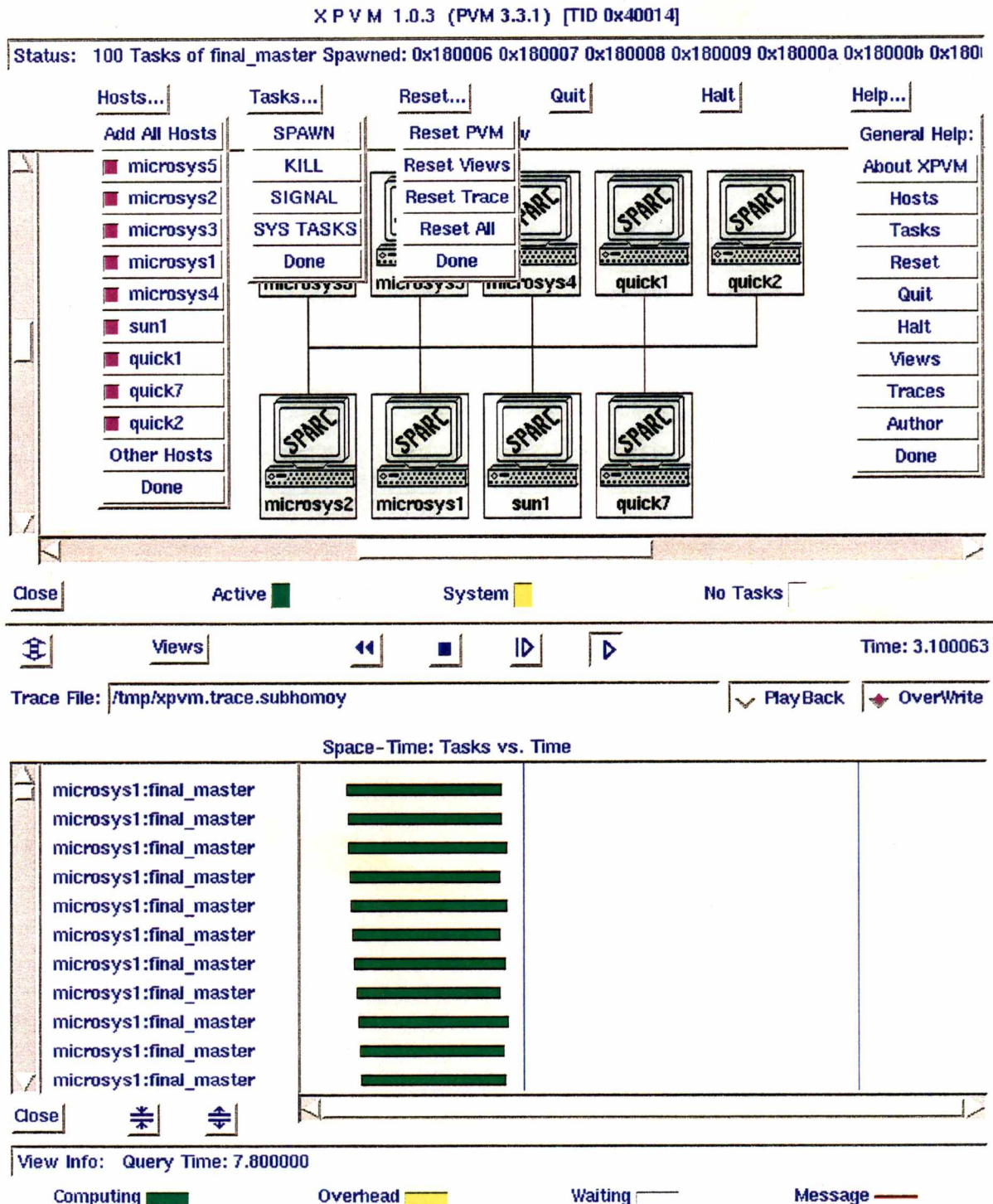


Figure 5.2: XPVM GUI with the Different Menus

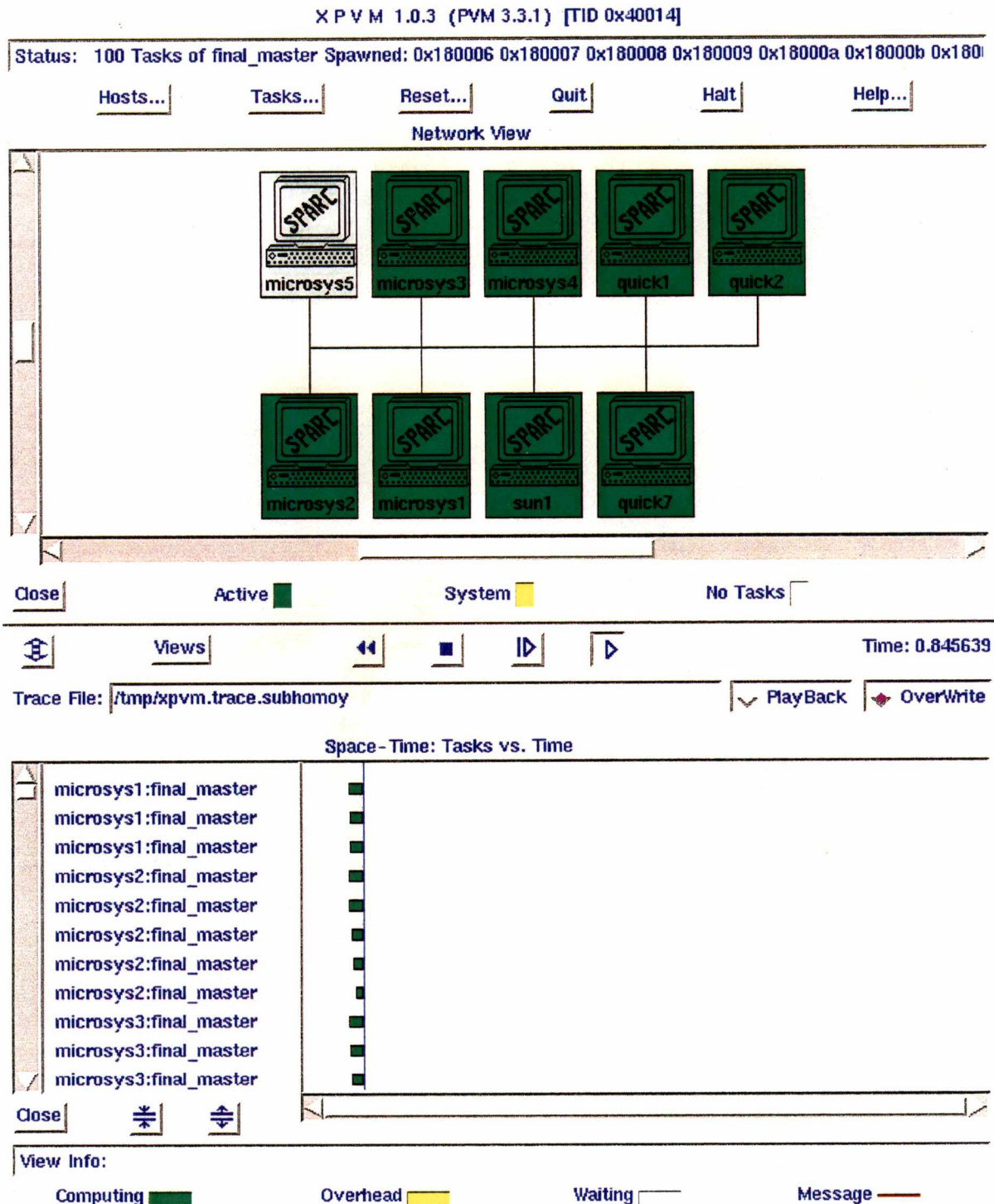


Figure 5.3: PVM Spawning Process Running on Different Machines

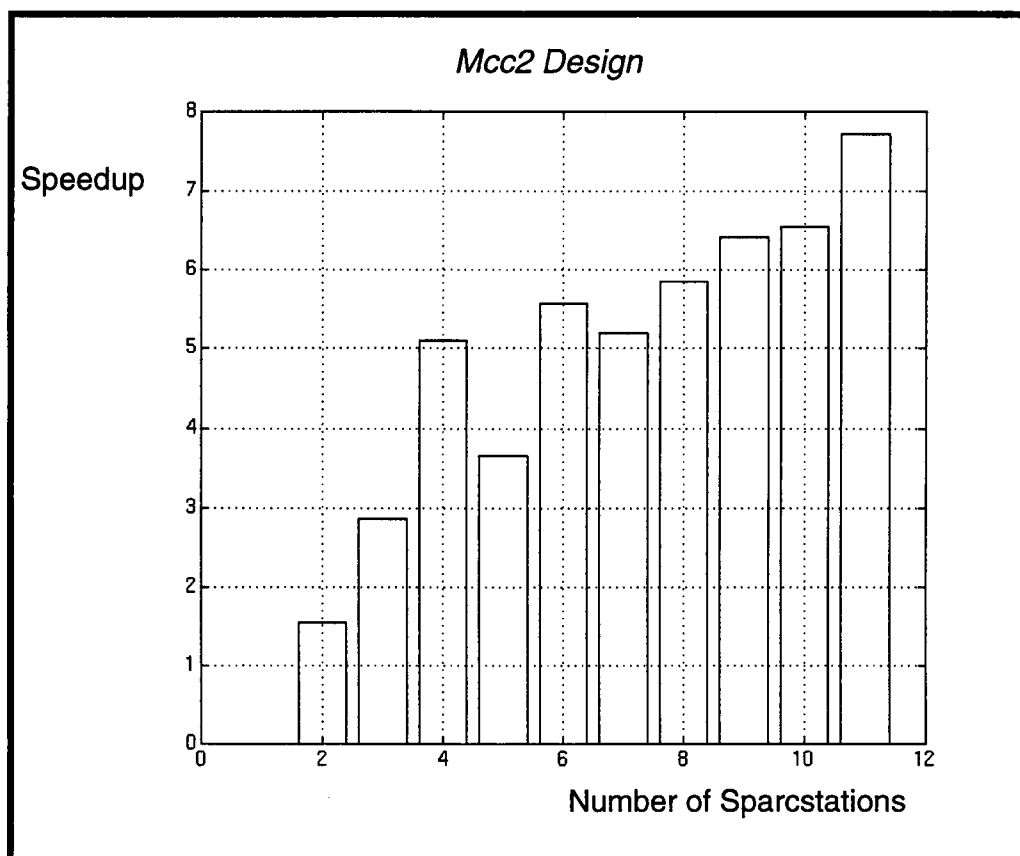


Figure 5.4: *Speedup Obtained for the Mcc2 Design (37 dies)*

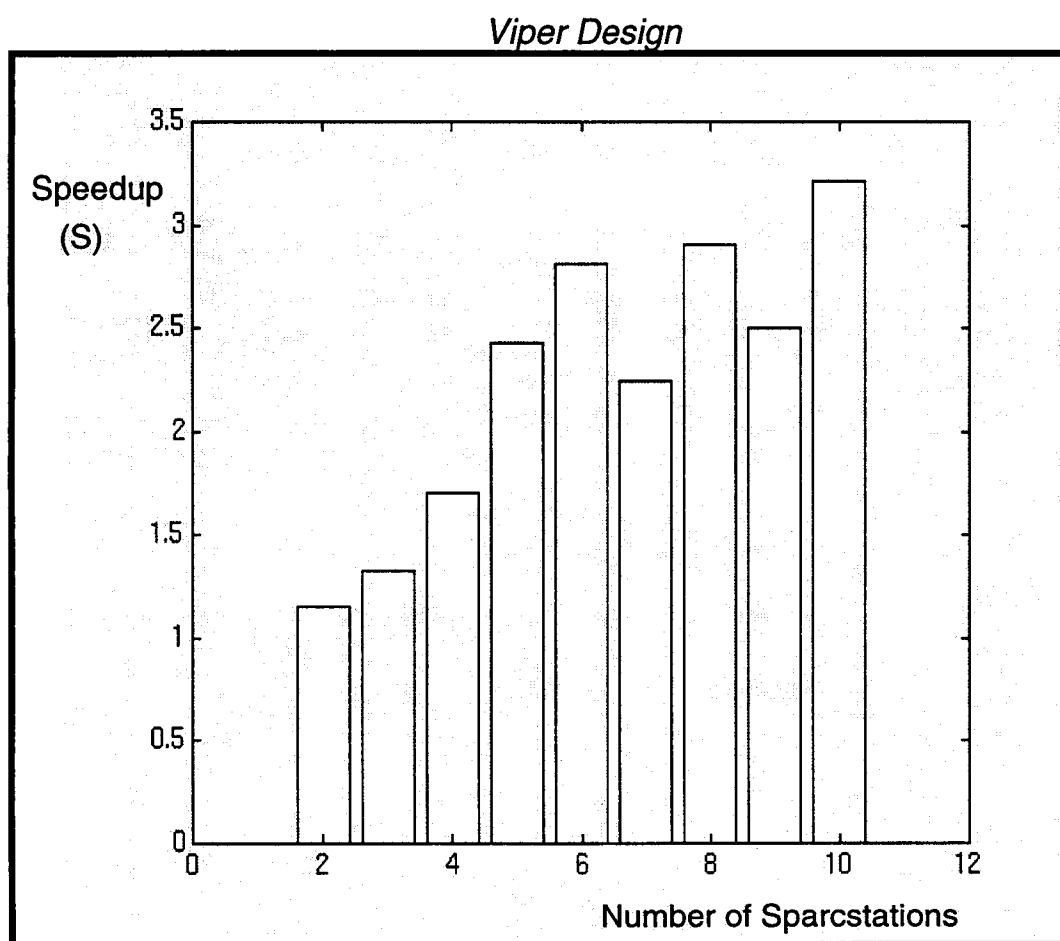


Figure 5.5: *Speedup Obtained for the Viper Design (10 dies*

We can see by looking at the figures that the speedup obtained for the larger example (37 dies, 7118 nets) is much larger compared to Viper example. This justifies the use of a distributed computing environment like the PVM even more for larger designs.

Table 5.1 and Table 5.2 show the improvements obtained by the PVM implementation for the MCM placement optimization program. We have run the program using up to 12 Sparc stations in a network. There is a substantial improvement in computation time for large designs containing a large number of nets when we run the optimization in a PVM environment. The factor of improvement in computation time for the design with 37 dies is close to *eight*. The improvement for smaller designs is also substantial as can be seen from the tables below. The speedup can be even more when the workstations are heavily loaded with a lot of memory and CPU intensive processes running.

5.4 Comments and Conclusions

In this chapter the results of a PVM implementation of a simulated annealing algorithm for placement of dies on the MCM substrate is presented. There is a substantial improvement in the computational time by using this approach especially in the case of placement of dies for a large design with a large number of nets. The hardware overheads are minimal as in most laboratories a network of workstations is easily available. It can be shown that the improvement in speed beyond a certain number of workstations is not substantial. An improvement

Table 5.1: *Run Times and Speedup for MCM Placement of Mcc2 in a PVM*

# Sparcs	Initial Cost	Final Cost	Run Time (s)	Speed Up	Efficiency
1	1.75847e+09	4.2581e+06	5930	1.00	1.00
2	3.67616e+09	1.5490e+07	3841	1.54	0.77
3	2.2287e+09	1.09387e+07	2072	2.86	0.95
4	9.9891e+08	2.17297e+07	1159	5.11	1.27
5	1.4543e+09	1.59833e+06	1623	3.65	0.73
6	1.8835e+09	4.93231e+06	1064	5.57	0.93
7	3.0408e+09	1.7193e+07	1144	5.20	0.74
8	4.0227e+09	5.3384e+06	1012	5.85	0.74
9	3.6047e+09	1.3531e+07	923	6.42	0.67
10	7.75069e+07	3.0094e+06	903	6.56	0.65
11	1.1518e+09	1.3689e+07	810	7.72	0.67

to this approach would be to implement both the cost function calculation as well as the optimization on different machines. Presently only the cost function calculation is done on several workstations as *slave* processes. Also instead of calculation of the overall cost each time there can be saving in computational time by calculating the incremental cost every time.

Table 5.2: *Run Times and Speedup for MCM Placement of a 10 die, Viper Microprocessor in a PVM*

# of Sparcs	Initial Cost	Final Cost	Run Time (secs)	Speed Up
1	3.86077e+07	1.96836e+07	90	1.00
2	2.64341e+07	1.75155e+07	78	1.15
3	3.21721e+07	1.68708e+07	68	1.32
4	3.32675e+07	1.79811e+07	53	1.70
5	2.67995e+07	1.79497e+07	37	2.43
6	3.22213e+07	1.92566e+07	32	2.81
7	3.29111e+07	1.75243e+07	40	2.25
8	3.56653e+07	1.82969e+07	31	2.90
9	3.51394e+07	1.93769e+07	36	2.5
10	2.92333e+07	2.08008e+07	28	3.21

CHAPTER 6

SUMMARY, CONCLUSION AND FUTURE WORK

6.1 Summary and Conclusion

The packaging bottleneck has been a great concern to VLSI designers in recent years. The role of packaging has grown dramatically in importance. Multichip modules have offered a partial solution to the packaging problem. The need and usefulness of MCMs as well as the different MCM technologies have been introduced in Chapters 1 and 2. The different commercial design methodologies for MCMs have been introduced in Chapter 4. Chapter 4 also dealt with the current design flow for MCM design. We have also pointed out the need for a new automatic approach to the placement of dies on the MCM substrate considering the netlength minimization and the thermal criteria simultaneously. This approach has been described in detail in Chapter 2. We can see from the placement results that the netlength increases by a small percentage when we consider both the thermal and the netlength criteria for optimization and the heat penalty is much lower indicating a more uniform thermal distribution. This is a very important result which needs to be further verified through the use of more accurate thermal models. In Chapter 5 we have reformulated the problem to run the simulated annealing based optimization algorithm in a Parallel Virtual Machine

(PVM) environment. It has been observed that very impressive speedups have been obtained especially for the larger designs when using multiple workstations connected in parallel. The maximum speedup obtained for the *mcc2* example with 37 dies, was 7.72. This was obtained using 11 workstations in parallel in a PVM environment. The use of a parallel environment for optimization is useful because it allows the use of more complicated heuristics and better thermal schedules resulting in better quality solutions for the same runtime as compared to a single machine environment. This is another advantage of using a distributed computing environment like PVM.

Thus by considering multiple criteria simultaneously for the placement and execution the optimization routines in a PVM environment we get a slightly higher total netlength (meaning delay and area) but considerably lower heat penalty and much shorter execution time especially for the larger designs.

The tool takes input designs in the MCC format and generates the output in the same format. A layout is also available showing the footprints of the dies and their locations on the MCM substrate. A XPVM X-window display environment can be used for adding and deleting hosts from the PVM, spawning processes and showing the activities on the different machines.

6.2 Future Work

As mentioned earlier we can conclude that thermal management in MCMs is an important issue. More accurate thermal models have to be considered which

will include the substrate conduction characteristics, convection coefficient of the air and use of heatsinks. We can make the placement tool for the components on the MCM substrate more complete this way.

We also have to consider delay constraints when doing the placement. A combined partitioning, placement approach may be the solution. We also need to find out a way of estimating the weights used in the cost function more accurately and convert all the different costs to a dollar value.

As of now the cost function calculation part in the simulated annealing based optimization procedure is being calculated in parallel while the actual optimization part is being done on the host machine. It would be better to have the cost function calculation as well as the optimization both execute in parallel. Also adaptive thermal profiles that are better but computation intensive can be tried when using the PVM approach.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] K.A. Sakallah A.L. Kayassi and Richard Brown. Impact of MCMs on System Performance Optimization. In *International Symposium on Circuits and Systems, (ISCAS)*, pages 919-921, 1992.
- [2] J. D. Cho and M. Sarrafzadeh. The Pin Redistribution Problem in Multichip Modules. In *Mathematical Programming, North Holland - Special Issue on Applications of Discrete Programming in Computer Science*, pages 297-330, 1994.
- [3] Mentor Graphics Corporation. *Mentor Graphics MCM Station Overview*. Mentor Graphics Corporation, Wilsonville, OR, 1993.
- [4] Peyman H. Dehkordi and Donald W. Bouldin. Design for Packageability - Early Considerations of Packaging from a VLSI Designer's Viewpoint. In *IEEE Computer*, pages 76-81, April 1993.
- [5] Giriraj Devaraj. *Distributed Placement and Crosstalk Driven Router for Multichip Modules*. MS Thesis, University of Cincinnati, 1994.
- [6] Schweikert D.G. A 2 Dimensional placement algorithm for the layout of electrical circuits. In *Proc. of 13th. Design Automation Conference*, pages 408-416, 1976.
- [7] D.A. Doane and P.D. Franzon, editors. *Multichip Module Technologies and Alternatives: The Basics*. Van Nostrand Reinhold (New York), 1992.
- [8] Naveed Sherwani et. al. *Routing in the Third Dimension From VLSI Chips to MCM*. IEEE Press New York, 1994.
- [9] P.D. Franzon. MCM package selection: A system's need perspective. In D.A. Doane and P.D. Franzon (ed), editors, *Multichip Module Technologies and Alternatives: The Basics*, chapter 3. Van Nostrand Reinhold (New York), 1992.
- [10] Cadence Design Systems Inc. A Vision for Multichip Module Design in the Nineties. Technical report, Cadence Design Systems Inc., Santa Clara, CA, 1993.
- [11] Oak Ridge National Laboratory. PVM - 3 Users guide and Reference Manual. Technical report, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1991.
- [12] Thomas Lengauer. *Combinatorial Algorithms for IC Layout*. Applicable Theory in Computer Science, Wiley Teubner Series, 1990.

- [13] Garay M. and D. Johnson. *Computers and Interactability: A guide to the theory of NP completeness*. Freeman, San Francisco, 1979.
- [14] Peter A. Sandborn and Hector Moreno. *Conceptual Design of Multichip Modules and Systems*. Kluwer Academic Publishers, 1993.
- [15] M. Sriram and S.M. Kang. *Physical Design of Multichip Modules*. Kluwer Academic Publishers, 1993.
- [16] D Bouldin Subhomoy Chattopadhyay, M. Sriram and Peyman Dehkordi. An Overview of Placement and Routing Algorithms for Multi-Chip Modules. In *Special Issue on MCM - International Journal of High Speed Electronics and Systems*, page Under Review, December 1995.
- [17] Barry Tolnas. *Multi Objective Design Advisor (MODA)*. MS Thesis, University of Tennessee, Knoxville, 1995.
- [18] Rao R. Tummala. Multichip packaging - A tutorial. In *Proceedings of IEEE*, pages 1924-1942, 1992.
- [19] Ram Vemuri. *Genetic Algorithms for Partitioning Placement and Layer Assignment for Multichip Modules*. Ph.D Dissertation, University of Cincinnati, 1994.

APPENDIX

APPENDIX A

AUTOROUTER ROUTING REPORT FROM MCM STATION FOR VIPER DESIGN

ELAPSED cpu= 0:00:03 clk= 0:00:04 -- Data Input

ELAPSED cpu= 0:00:05 clk= 0:00:07 -- Image Definition

| Date and Time Report

Wednesday, June 28, 1995 06:10:09 pm

| Units Used Report

Units of length are in inches.

| Memory Usage Report

Expected total of 6.345 megabytes of memory required.

Layer Report

Layer	Name	Layer Type	Direction	Map Built
1	bond	Bond	No Routing	No
2	crossover	Signal	Vertical	Yes
3	discrete_diel	Dielectric	No Routing	No
4	conductor_1	Signal	Horizontal	Yes
5	sheet_diel_1	Dielectric	No Routing	No
6	conductor_2	Signal	Vertical	Yes
7	sheet_diel_2	Dielectric	No Routing	No
8	conductor_3	Signal	Horizontal	Yes
9	sheet_diel_3	Dielectric	No Routing	No
10	conductor_4	Signal	Vertical	Yes
11	base	Unknown	No Routing	No

Via Report

Via Name	Style	Layer	Shape	Size
----------	-------	-------	-------	------

4	Square	0.0050
6	Square	0.0050
8	Square	0.0050
10	Square	0.0050

Via		Layer Range		Via Site Count	
Rule	Via Name	Min	Max	Normal	Under Pad
1	5x5	4	10	71822	0
2	5x5	6	10	89401	0
3	5x5	4	8	71822	0
4	5x5	8	10	89401	0
5	5x5	6	8	89401	0
6	5x5	4	6	71822	0
7	5x5	2	4	71822	0
Totals				555491	0

| Design Rule Report

Net Type

Name	Layer	Width	P-V	P-T	V-V	V-T

DEFAULT_NET_TYPE	2	0.0010	0.0040	0.0010	0.0040	0.0010
	4	0.0010	0.0040	0.0010	0.0040	0.0010
	6	0.0010	0.0040	0.0010	0.0040	0.0010
	8	0.0010	0.0040	0.0010	0.0040	0.0010
	10	0.0010	0.0040	0.0010	0.0040	0.0010

Net Type

Name	Layers and Autorouter Vias

DEFAULT_NET_TYPE	Layers: bond, crossover, conductor_1, conductor_2 conductor_3, conductor_4 Vias: 5x5

| Connection Report

Net Type

Name	Total	Selected	Protected	Routed
------	-------	----------	-----------	--------

DEFAULT_NET_TYPE 696 696 0 0

Pin Report

Pin Type Total Off-grid Unused

top surface 1251 1206 6

Pins per Net Net Count

2 476

3 34

4 34

8 3

11 1

20 1

Routing Controls Report

Routing Controls	Value

breakout passes	0
pattern passes	0
automatic passes	5
manufacturing passes	0
intersections	Removed
sort order	Ascending Length
balance layers	Off
t-junctions	On
diagonal routing	On
diagonal only bends	Off
diagonal only pin exit	Off
fence units	0.100

Outline Type	Min X	Min Y	Max X	Max Y

routing	-1.5000	-1.5000	1.5000	1.5000

%-----

%| Routing Grid Report

%-----

%	Grid	X	Y
---	------	---	---

%-----

%	trace	0.0250	0.0250
---	-------	--------	--------

%	pin	0.0000	0.0000
---	-----	--------	--------

%	via	0.0100	0.0100
---	-----	--------	--------

%-----

%Grid Type	X Grids	Y Grids
------------	---------	---------

%-----

%Non-uniform	359	359
--------------	-----	-----

%-----

| Automatic Routing Summary

Inter- Completion

Pass	Schedule	Unroutes	Vias	sections	Selected	CPU	Clock
------	----------	----------	------	----------	----------	-----	-------

1	0	695	0	0	0.1%	0:00:03	0:00:05
2	1	209	1741	807	69.9%	0:45:57	0:52:59
3	2	99	1782	151	85.7%	0:36:55	0:37:22
4	3	48	1765	63	93.1%	0:34:16	0:34:41
5	16	13	1765	15	98.1%	0:05:08	0:05:11

Routing Statistics Report

Layer	Paths	Horz	Vert	Diag	Total
2	264	4.52	111.78	1.61	117.91
4	1991	222.88	13.94	6.60	243.42
6	615	3.39	220.87	2.76	227.03
8	672	377.06	1.85	2.72	381.63
10	161	3.03	86.27	1.63	90.93

 Totals 3703 610.87 434.72 15.34 1060.93

Manhattan Length	Completed Manhattan	Percent Manhattan	Routed to Manhattan Length Ratio
---------------------	------------------------	----------------------	-------------------------------------

 1034.13 1015.68 98 1.045

Total Vias	Power Vias	Ratio of Vias Connections to Connections
---------------	---------------	---

 1743 0 696 2.504

Via Name	Rule	Zlo	Zhi	Count
----------	------	-----	-----	-------

5x5	1	4	10	113
5x5	2	6	10	4
5x5	3	4	8	309
5x5	4	8	10	91
5x5	5	6	8	351
5x5	6	4	6	496
5x5	7	2	4	379

ELAPSED cpu= 2:03:22 clk= 2:11:25 -- Automatic Routing

| Routing Controls Report

Routing Controls	Value
------------------	-------

breakout passes	0
-----------------	---

pattern passes	1
----------------	---

automatic passes	0
------------------	---

manufacturing passes	0
----------------------	---

intersections	Removed
sort order	Ascending Length
balance layers	Off
t-junctions	On
diagonal routing	On
diagonal only bends	Off
diagonal only pin exit	Off
fence units	0.100

	Inter-	Completion Rate
Unroutes	Vias	sections Selected Overall
13	1765	0 98.1% 98.1%

Automatic Routing (Pattern) -----

ELAPSED cpu= 0:00:03 clk= 0:00:04 -- Routing Pass Execution

| Routing Statistics Report

Layer	Paths	Horz	Vert	Diag	Total
-------	-------	------	------	------	-------

2	264	4.52	111.78	1.61	117.91
4	1991	222.88	13.94	6.60	243.42
6	615	3.39	220.87	2.76	227.03
8	672	377.06	1.85	2.72	381.63
10	161	3.03	86.27	1.63	90.93

Totals	3703	610.87	434.72	15.34	1060.93

Manhattan Length	Completed Manhattan	Percent Manhattan	Routed to Manhattan Length Ratio

1034.13	1015.68	98	1.045

Total Vias	Power Vias	Ratio of Vias Connections to Connections

1743	0	696 2.504

Via Name	Rule	Zlo	Zhi	Count

5x5	1	4	10	113
5x5	2	6	10	4

5x5	3	4	8	309
5x5	4	8	10	91
5x5	5	6	8	351
5x5	6	4	6	496
5x5	7	2	4	379

VITA

Subhomoy Chattopadhyay received the Bachelor of Science degree in Electrical Engineering from University of Calcutta, India in 1989. He obtained his MTECH degree from Indian Institute of Technology, Madras, India in Electrical Engineering in 1991. His area of specialization for his MTECH degree was in Power Electronics and Controls. During his MTECH he worked as a Research Assistant and later as a Scientific Officer Grade II in a Government of India funded project in the area of Simulation of Power Electronics and Control Systems. He worked with the Indian industries to develop power electronic controllers. He came to University of Tennessee in 1992 for higher studies. From January 1992 to August 1993 he worked as a Graduate Research Assistant at EPRI, Power Electronics Applications Center (PEAC) in Knoxville, Tennessee. Later he worked for a year as a Graduate Assistant at the University of Tennessee Computing Center, Knoxville, TN. His work at UTCC involved user support on UNIX and mathematical programming languages. He also taught short courses to University of Tennessee students and faculty. From September 1994 he has been working as a Graduate Research Assistant at the Microelectronics Systems Research laboratory, University of Tennessee, Knoxville. He is presently working on the ARPA funded project on Design for Packageability.