Doctoral Thesis

System design using high density packaging and multi chipmodules

Author(s): Hirt-Schnurrenberger, Etienne Ramon

Publication Date: 2000

Permanent Link: https://doi.org/10.3929/ethz-a-003889863

Rights / License: In Copyright - Non-Commercial Use Permitted
System Design using High Density Packaging and Multi Chip Modules

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH
for the degree of
Doctor of Technical Sciences

presented by
Etienne Ramon Hirt-Schnurrenberger
Dipl. El.-Ing. ETH
born October 5th, 1970
citizen of Brugg, AG

accepted on the recommendation of
Prof. Dr. Gerhard Tröster, examiner
Prof. Dr. Paul Franzon, co-examiner

March 2000
Biography

Etienne Hirt received a Dipl. Ing. (M.S.) degree in electrical engineering from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, in 1995. He joined the Electronics Laboratory of the ETH in spring 1995 as a teaching and research assistant in the Multi Module Group. He worked in the fields of system and package design methodology using Multi Chip Modules and High Density Packaging technologies. In 2000, he received the Dr. sc. techn. (Ph.D.) degree from the ETH Zurich.

He jointly developed two Pentium based multi chip modules and an FPGA based MCM and he is co-founder of the ETH Spin-off company Art of Technology AG.
Abstract

High Density Packaging (HDP) and Multi Chip Modules (MCM) enable designers to address the industry's requirements such as increasing processing power packaged in smaller and smaller form factors, using less power and with easy thermal management. HDP and MCM technologies can

- reduce cost at system level,
- increase functionality while decreasing size and weight,
- maximise performance of the applied ICs,
- reduce power consumption,
- increase reliability,
- bolster protection against EMC and EMI,
- and increase modularity and reusability of subsystems.

However, HDP stands for a bunch of technologies. The ICs are either used as bare dies, attached with wirebond, TAB or flip chip, or are packaged into standard or advanced packages. The ICs are then mounted onto laminate, ceramic or thinfilm substrates depending on the system requirements. Additionally to this technology selection, a design has to be partitioned into modules featuring the same or compatible technologies thus setting up a design space with many degrees of freedom.

To make HDP and MCM design tractable, a new design flow was developed. It includes a feasibility analysis where prior to layout, technical alternatives are created and analyzed. This design space exploration is done in a mixture of top-down and bottom-up manner and allows a cost-performance tradeoff to choose one option for implementation and layout.

To improve and automate the proposed feasibility analysis, new models were developed that estimate substrate size as well as the necessary layers count. The lower limit of a substrate (the chip limited size) is dominated by the component’s footprints. In this thesis the footprint is defined as the pure component area and the area consumed for the interconnect to the substrate and the escaping from the substrate pads or the assembly keepout. This footprint therefore depends on the first level interconnect/package and the substrate type as modeled in this thesis.

Often the substrate size is defined by the interconnect requirements. To calculate the cost of an implementation, a designer has to know how large as substrate will be for different layer counts. Furthermore, the space between the components is important. To address these requirements, a size estimation method was developed using global routing. The method considers escape routing as well as interconnection under the components and minimizes the substrate size for a given number of layers by spreading the space between two components as little as needed. Additionally, the method supports routing constraints such as maximum wire lengths, fixed topologies or keepout areas.

The developed methodologies and models are explained and illustrated, and their accuracy is proven with case studies. Furthermore, they were the basis for the implementation of in the “JavaCAD” tool.
Zusammenfassung

In der Mikroelektronik tätige Unternehmen führen einen Wettlauf gegen die Zeit - im Bestreben, ständig kleinere, leichtere und kostengünstigere Produkte auf den Markt zu bringen. High Density Packaging (HDP) und Multi Chip Module (MCM) bieten ihnen neue Möglichkeiten zur Verbesserung ihrer Wettbewerbsfähigkeit. HDP und MCM Technologien ermöglichen

- Reduktion der Kosten auf Systemebene,
- Erhöhung der Funktionalität bei reduzierter Grösse und verringertem Gewicht,
- Ausnutzung der vollen Leistungsfähigkeit der eingesetzten ICs,
- Reduktion der Leistungsaufnahme,
- erhöhte Zuverlässigkeit,
- einfacherer EMV- und ESD-Schutz,
- erhöhte Modularität und Wiederverwendbarkeit der Baugruppen.


Um die vorgeschlagene Machbarkeitsanalyse zu vereinfachen und zu automatisieren wurden neue Modelle entwickelt, die die Substratgrösse und die benötigte Anzahl Lagen bestimmen. Die minimale Substratgrösse wird durch die sogenannten Footprints der Komponenten bestimmt. In dieser Arbeit sind die Footprints definiert als die Komponentengrösse plus die Fläche die benötigt wird, um die Komponenten mit dem Substrat zu verbinden, plus dem Wegrouten der Pads zu Vias oder der Freihaltelefläche für die Montage. Dieser Footprint ist also abhängig von der Verbindungstechnologie zwischen dem IC und dem Substrat und der Substratechnologie. Er wird in dieser Arbeit beschrieben.

Oft ist aber die Substratgrösse durch die benötigte Verbindungsdichte bestimmt. Um die Kosten einer Implementation berechnen zu können, muss ein Designer die Substratfläche in Abhängigkeit von der Lagenanzahl kennen. Zudem sind die benötigten Abstände zwischen den Komponenten wichtig. Um diese Anforderungen abzudecken, wurde eine Grössenabschätzung mit Hilfe von Global Routing entwickelt. Die Methode berücksichtigt das Wegrouten der Komponentenanschlüsse zu Vias und die Verbindungen unter den Komponenten. Sie speizt die Abstände zwischen den Komponenten so wenig wie möglich und minimiert damit die Substratgrösse für
eine vorgegebene Anzahl Substratlagen. Zudem kann diese Methode weitere Anforderungen, wie maximale Leitungslängen, Leitungstopologien oder Sperrflächen, berücksichtigen.

Acknowledgement

I would like to express my gratitude to my advisor Prof. Dr. G. Tröster for giving me a very high degree of freedom in my research. I would like to thank my associate advisor Prof. Dr. P. Franzon for co-examining and for his valuable inputs to this thesis.

Thanks to all the members of the multi chip module research group. Together we built-up the basic know how for this work. Special thanks go to Jean-Pierre Wyss for his excellent and fruitful co-operation in demonstrator designs and in the Europractice project, and to Michael Scheffler for his valuable input to the development of the JavaCAD tool as well as comments on papers and the thesis. The fruitful discussions during coffee-breaks and other occasions with members of the lab were always helpful.

Thanks to the students for their contributions during their graduate work.

Finally, I want to thank my wife and best friend, Evi. Without her help and support this work would not have been possible.

Zurich, March 2000

Etienne Hirt
Contents

1. Introduction
   1.1. Our Solution, the MCM and HDP Technologies
   1.1.1. Advantages
   1.1.2. GPS MCM
   1.1.3. Chip-Package Co-Design
   1.1.4. Challenges
   1.2. Motivation
   1.2.1. Questions to be answered
   1.2.2. Existing Approaches
   1.3. Novel Contributions
   1.4. Structure

2. Technologies
   2.1. First Level Interconnects
      2.1.1. Wire Bonding
      2.1.2. TAB
      2.1.3. Flip Chip (FC)
      2.1.4. Chip Size Package
      2.1.5. Comparison of First Level Interconnects
   2.2. Substrate Technologies
      2.2.1. Laminate
      2.2.2. Sequential Build-Up
      2.2.3. Thin Film (MCM-D)
      2.2.4. Hybrid
      2.2.5. Cofired Ceramic
      2.2.6. Ceramic Photodefinable
      2.2.7. Comparison of Substrate Technologies
   2.3. Integrated Passives
      2.3.1. Integrated Resistors
      2.3.2. Integrated Capacitors
      2.3.3. Integrated Inductors
   2.4. Summary

3. Design Space Exploration
   3.1. Design Flow
   3.2. Procedure
   3.3. The Pentium MCM
      3.3.1. Design Goals
   3.4. Partitioning
      3.4.1. Partitioning Functionality into Chips
      3.4.2. Component Set Partitioning and Assigning Partitions to Modules
      3.4.3. Partitioning of the Pentium MCM
   3.5. Bonding Options
      3.5.1. Wire Bond
7.1.6. Final Selection ................................................. 116
7.2. GPS MCM .................................................. 117
    7.2.1. System Advantages and actual Implementation .......... 118
    7.2.2. Bonding Options and Footprints ......................... 118
    7.2.3. Chip Limited Size ...................................... 120
    7.2.4. Routing Limited Size ................................... 120
    7.2.5. Cost/yield analysis .................................... 124
    7.2.6. Conclusion .............................................. 125
7.3. Summary ..................................................... 125

8. Summary and Conclusions ........................................ 127
    8.1. Design Space Exploration .................................. 127
    8.2. Substrate Area and Layer Count Estimation ................ 128
    8.3. JavaCAD ................................................... 128
    8.4. Limitations ............................................... 129
    8.5. Impact of the NTRS Roadmap on Substrate and Assembly Technologies ............... 129
    8.6. Future Work ............................................... 130

A. Utilized Parameters and their Symbols ........................ 133
    A.1. Chip/Package Parameters ................................. 134
    A.2. Assembly Parameters ...................................... 135
    A.3. Substrate Parameters ...................................... 135
    A.4. Calculated Parameters .................................... 136
Introduction

The rapidly growing market for embedded and portable computing as well as mobile equipment exhibit a steadily increasing demand for ever increasing processing power packaged in smaller and smaller form factors, using less power and, from the users perspective, easy thermal management. In addition, systems are required to exhibit ever increased reliability. All this has to be achieved at lower cost. Area can be reduced and speed enlarged through greater integration, but this can be taken only so far. In order to minimize the overall system area, the printed circuit board (PCB) area has to be minimized.

Actually, the above needs can not be fulfilled by standard SMT solutions because of the following: An IC-package is much bigger than the ICs itself due to the high number of interconnections to the outside. The packages also rend the signals between the ICs much slower because they add distance and heavy load to the lines which both enlarge the time of flight heavily. These parasitic loads are also responsible for the high power consumption of output drivers, because they do not only have to drive the input of the other ICs but also the parasitics of the physical interconnection. The slow signal interconnections and the moderate number of IOs available in a package lead to big IC designs which suffer a small yield, what rends them more expensive. Furthermore, the big packages do not only influence the speed but they also lead to the necessity of partitioning the design onto several boards. These separate boards add even more to the cost.

The answer to these challenges are advanced packaging options such as multichip modules (MCMs) and high density packaging (HDP) as described in the next section. However, it is very difficult to choose the suitable technology if one is not very experienced in each of the many technologies. Therefore, methods and models have to be developed to manage the many degrees of freedom.

1.1. Our Solution, the MCM and HDP Technologies

An MCM is an electronic system or subsystem with two or more bare integrated circuits (bare die) or Chip Sized Packages (CSP) assembled on a substrate. The substrate is either a PCB, a thick or thin film ceramic or silicon with an interconnection pattern. The substrate is either an integral part of a package or will be assembled into a package. Thus, several components are combined to a single “component” or a complete product requiring a much smaller area than the original packaged components with much less I/Os to connect.

This leads to the advantages presented below.
1.1.1. Advantages

Multichip Modules introduce a packaging level between ASICs and PCBs other than Single Chip Package forming an electrically, thermally and mechanically optimized subsystem. There are many reasons why this might be beneficial. The driving forces to develop an MCM instead of using standard packaged circuits are:

- **Size**: The utilisation of the active silicon area is about 15% for surface mounted circuits on a PCB. In an MCM it can be between 30-60% or even higher.

- **Complexity**: A large system can be divided into several very dense circuits. The dimensions of an IC mounted on an HDP substrate (footprints) are close to the dimensions of the ICs themselves. So, high complexity circuits can be assembled without large fan out and interconnections can be made with fewer layers in the holding motherboard.

- **Technology Encapsulation**: In an MCM, digital and analog functions can be mixed without serious limitations and microcontrollers can be enhanced by memory without enlarging the CPU chip. Furthermore, passive components can be integrated for decoupling, protection and high precision analog functions. The next generation of MCMs may even be available with optical IOs.

- **Electromagnetic Compatibility, Signal Speed and Signal Integrity**: High speed components can be placed closer together, the load on the output buffer is lower and signal transmission properties are better. Furthermore, electromagnetic interference and compatibility are improved compared to a PCB.

- **Simplification of Motherboard Design**: Subsystems with high wiring demands can be integrated into an MCM with a limited number of external connections. So, the high wiring density is concentrated on a small area thereby relaxing the wiring pitch as well as the number of layers on the much larger motherboard.

- **Reusability/Standardisation**: An MCM allows to integrate functions required in a family of products as a component that can be handeld like a conventional packaged IC. These modules can be integrated as logical and physical building blocks in future designs.

- **Harsh Environment**: A small system can more easily be protected from electromagnetic interference, liquids, gases etc. than a larger system.

- **Cost**: Many low cost products are produced in large volumes as a Chip on Board or relatively simple MCMs in watches, calculators and in new products like video cameras and industrial PCs. Presently, however, the cost are generally equal or higher than for a PCB. Nevertheless, the overall system cost decrease when the module is manufactured in medium or high volumes due to the simplification of the motherboard design. So, integrating the same functionality without daughterboards lowers the cost even when using a more expensive "component".

Most of these advantages are illustrated with the GPS receiver module of the Swiss company µblox.

1.1.2. GPS MCM

The GPSMS1 of µblox AG [1] detailed in section 7.2 has the following advantages:
1.1. Our Solution, the MCM and HDP Technologies

- Size reduction by factor 3 shown in figure 1.1
- Interface simplified to RF input and serial output
- RF design encapsulated
- Can be used in many applications such as:
  - Smart antenna
  - Parcel or messenger pigeon tracking
  - Car navigation
  - Handheld navigation
- Marginally higher cost than the SMT counterpart

HDP technologies are also a solution for chip designers when they are ready to co-operate with system designers. Such concurrent development is known as chip-package co-design.

1.1.3. Chip-Package Co-Design

The HDP technologies can also be used to split large IC designs into several smaller ones to improve the overall yield of a component [3][11]. Besides a low yield large ICs also suffer from long on-chip interconnections. Due to their high resistivity and high capacitive load they become slow. This effect was named $\frac{1}{3}$ by E. Davidson[1]. The speed degradation can be overcome by using low loss off chip-routing as shown in figure 1.2. But still the semiconductor industry forces the single chip design. The increasingly higher integration of transistors at much lower cost per transistor has resulted in a capability of placing more than 20 million transistors on a single chip. These facts enable IC manufacturer to build digital systems such as microprocessors with periphery and memory. Thus, semiconductor industry proposes system on a chip (SOC). Tummala[5] defines the SOC as the realization of an entire systems’s functionality in a single, large IC (with process compromises to accommodate various macros and technologies). This IC integrates digital, RF, analog, and other functions as shown in figure 1.3(a). The SOC are not only difficult to manufacture and to test but also a challenge to design because chip designers have to integrate and rely on designs done by other companies. To avoid the mentioned challenges and to profit from the advances in packaging technology Tummala proposes system on a package (SOP). An SOP, as illustrated in figure 1.3(b), is similar to an SOC in that it is a single component. In contrast to an SOC, however, it is a system-level package containing multiple large ICs (the largest ICs manufacturable at the
highest yield and hence lowest cost) that integrates all the system functions together with all the required passive components often integrated into the substrate. Although an SOP may not always achieve the performance and form factors of an SOC, in most cases it will be a quicker, low-cost, efficient, and low-risk solution.

The SOP paradigm proposes an unified chip-plus-package view of the design process. The required chip-package co-design approach [6] was proposed earlier and lead to already two workshops at ETH [7] [8]. An example for chip package co-design is a processor with first level cache off chip[9] as illustrated in figure 1.4. It profits from the faster off chip lines and the memory technology much denser than logic technology for the SRAMs. Additionally, the DRAM controller is implemented on the CPU chip ensuring low latency and allowing for two independent DRAM banks. With this architecture, the functionality of a CPU could be distributed freely on an MCM to achieve optimal performance and yield.

However, to profit from these advantages some challenges have to be managed.
1.2. MOTIVATION

1.1.4. Challenges

To profit from the above advantages some challenges have to be overcome. They range from partitioning, over all the physical aspects such as high density substrate selection and power dissipation concentration to testability and die supply:

- **Partitioning**: Which components should be integrated into an MCM?
- **Technology Selection**: Which interconnect can fulfill the needs?
- **Power Dissipation**: How to get rid of the power concentrated?
- **Test**: How to make an MCM fully testable as it combines the complexity of a fully assembled board with the access limitations of an IC?
- **Die Supply**: Are bare dies or CSPs available and how are they tested?

1.2. Motivation

To manage the challenges of MCM design, a good design methodology is required. The difficult task is not the physical design, but all the decisions that have to be taken prior to CAD work. As the interconnection influence strongly the system performance, weight and cost, all the physical aspects have to be considered early in design. Furthermore, a system design including MCM design cannot be a straightforward approach because it is not obvious from the beginning which combination of components and packaging solutions will be the optimal system. During the design process, many aspects as described below in subsection 1.2.1 have to be evaluated.

1.2.1. Questions to be answered

To manage the many degrees of freedom, the HDP design flow must be a mixture between top-down and bottom up-approach. It is top-down to choose the optimum build-up without performing any layout. But the design flow cannot neglect the technology details considered in the bottom up
approach. A design space exploration combines these two contrary approaches by enhancing the high level design with build-up details. In this design space exploration more and more alternative component sets and packaging options are generated and some of them are skipped as described in section 3. After setting up the alternative component sets and packaging options, a detailed feasibility study has to answer the following questions:

1. **Size**: Is a considered module or partition of reasonable size and does this lead to an optimal system size?

2. **Technology**: Which technologies suit the MCM requirements most?

3. **Yield**: Does the chosen implementation lead to an acceptable yield or how can we reach it?

4. **System**: What are the system advantages? Added value, smaller size etc.

5. **Cost**: Is this a cost effective system design? Can a daughterboard be saved, will the housing become much less expensive by using a more expensive "component" or are people ready to pay for the smallness or the added functionality? Thus, is the overall cost minimal?

Finally, when all the technical aspects have been considered, a cost-performance trade-off allows to choose the least expensive or the optimum alternative before starting physical layout. As the existing approaches described below in subsection 1.2.2 do not fulfil the above requirements, a new designflow and powerful models were developed as summarized in section 1.3.

### 1.2.2. Existing Approaches

Current approaches can be divided into two major directions. One type comes from the chip community [10, 11, 12]. It considers partitioning the functionality of one large IC into smaller ones using exclusively thin film substrates and flip-chip interconnect. The other approach from the packaging community [13, 14, 15, 16] includes more details from the package influences and proposes a concurrent approach already early in the design cycle.

While the first approach is limited to a high performance technology, the second lacks a proper design strategy as the proposed what-if analysis is very time consuming because it is a serial process. To speed-up the early design analysis, a methodology and models were developed as described in the next section.

### 1.3. Novel Contributions

To make HDP and MCM design tractable, a new design flow was developed. It includes a feasibility analysis where prior to layout, technical alternatives are created and analyzed. This design space exploration is done in a top-down manner and allows a cost-performance tradeoff to choose one option for implementation and layout.

To improve the proposed feasibility analysis, new models were developed that estimate substrate size as well as the necessary layers count. The lower limit of a substrate (the chip limited size) is dominated by the component's footprints. In this thesis the footprint is defined as the pure component area and the area consumed for the interconnect to the substrate and the escaping from the substrate pads or the assembly keepout. This footprint therefore depends on the first level interconnect/package and the substrate type. Often the substrate size is defined by the interconnect requirements. Overall routing estimations such as Seraphims estimation[17]. others as compared from Sandborn et al.[18] and Diaz-Alvarez [19] are not accurate enough. First, they are too optimistic as blocked areas from escaping are
neglected. Second, they estimate the overall substrate size, but cannot calculate the width of a channel between components necessary. To address these requirements, a size estimation method was developed using global routing. The method considers escape routing as well as interconnection under the components and minimizes the substrate size for a given number of layers by spreading the channels between two components as little as needed. Additionally, the method supports routing constraints such as maximum wire lengths, fixed topologies or keepout areas as described in chapter 5.

This leads to the structure described below.

1.4. Structure

This thesis first introduces the MCM technology in chapter 2. After this technical introduction the design space exploration as part of a high level design and feasibility study is explained in chapter 3. New models for footprint estimation and the chip limited size are introduced in chapter 4 and the routing estimation starting from this size is described in chapter 5. These models were implemented in a java based tool "JavaCAD" presented in chapter 6, and finally real design examples are described in chapter 7 to round off.
Technologies

The MCM and HDP technologies allow higher performance in terms of signal speed, smaller size and a reduction of external connections. The basic idea of MCMs is to decrease the average spacing between ICs in an electronic system. Therefore, the main target is chip interconnection technology. The MCM technology prefers using direct chip attach to single chip package and it interconnects the ICs on a high density substrate which is packaged or is a package itself. In spite of the common goals the implementation of an MCM leaves many options as presented in the following sections. The selection of the applied technologies has to be done carefully to find an optimum that fits to the system requirements. Generally speaking there are Rolls-Royce, Trabant or VW solutions for each system.

In the first section the first level interconnects that connect the active silicon with the outside are introduced. They are mainly the same as the ones used to connect ICs to a single chip package or are advanced packages. As often no standard packages are used the components interconnection requirements are seldom fulfilled by PCB technology due to its large routing pitch. An overview of suitable substrate technologies is given in section 2.2. Finally, another feature of these technologies, integrated passives, is described in section 2.3.

2.1. First Level Interconnects

First level interconnects are the mechanical and electrical interconnection from the die to the substrate[20]. The same technologies as for mounting ICs in single chip packages can be used. This interconnection can either be made by means of a wire (wire bonding), a foil (TAB), solder or adhesive (both for flip chip). Other options are to reroute the pads on-chip to provide area interconnect robust against die masks revisions, to use advanced packages or to have dies packaged into them. The advanced packages are ball grid arrays (BGA) and chip size packages (CSP) that distribute the pads over the whole package area. All the technologies are described in the above order in this section.

2.1.1. Wire Bonding

Wire bonding is the oldest and most mature technology used for bonding semiconductor chips. The chips are attached mechanically to the substrate by means of adhesive or solder. The selection of the attach material depends on electrical and thermal conductive requirements as well as reliability to act as an interposer if required. The pads of the chip are afterwards wirebonded to the substrate by aluminium or gold wires of typical 25 - 33 µm diameter as shown in figure 2.1. This technology minimizes risk because it is tolerant to small changes on chip and there are many
companies providing wire bonding. Furthermore, the yield is known as about 99.99% per bond or better, thus cost is predictable. Additionally, most ICs are made for wire bonding and a faulty wirebond can be reworked. But, wirebonding is a sequential process and the wires add significant inductance to a signal path.

There are three wire bonding methods used in the industry:

- thermocompression
- ultrasonic
- thermosonic

The thermocompression or thermosonic method is used for gold ball-wedge or wedge-wedge bonding, and ultrasonic bonding fits to aluminium wedge-wedge bonding.
2.1. First Level Interconnects

2.1.1. Wedge-Wedge Bonding

Wedge-Wedge bonding is used for both ultrasonic aluminium wire bonding and thermosonic gold bonding. Figure 2.3 shows the stages of the bonding process [20]. The ultrasonic energy is applied to a wire, held in contact with a bonding pad by the wedge or bonding tool. The combination of pressure and ultrasonic energy forms a metallurgical weld without addition of significant thermal energy for aluminium wire bonding as shown in figure 2.2(a). Gold wire bonding is not processed at ambient temperature. Instead the substrate has to be heated to 150°C. Therefore, aluminium wedge bonding is preferred to gold bonding because of the lower process cycle time and the avoidance of thermal stress due to excessive heating. Furthermore, the requirements of the substrate are much lower for aluminium: typically 4-5 μm nickel and flashgold (chemical 0.01-0.09 μm) provide good results while for gold wire bonding a gold thickness of 0.5 to 0.7 μm is required on the nickel.

Wedge-wedge bonding is well suited for small pitches mainly for the second bond which usually is on the substrate. However, it is slower than the ball-wedge method. If the wires do not leave the chip orthogonally the bond head for wedge-wedge bonding has to be turned for each bond which significantly slows the machine down. To get an idea, table 2.1 shows typical substrate bond pad sizes specified by manufacturers. The bond pad is longer than the length required for a single bond. This length is required to place a second bond in case of rework. When the bond pads are testpoints as well, they should be about 50 μm longer.

Table 2.1: Wedge-Wedge Substrate Bond Pad Sizes

<table>
<thead>
<tr>
<th>Substrate Type</th>
<th>minimum</th>
<th>optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCM-L</td>
<td>100*250</td>
<td>150*300</td>
</tr>
<tr>
<td>MCM-C</td>
<td>125*250</td>
<td>200*300</td>
</tr>
<tr>
<td>MCM-D</td>
<td>90*250</td>
<td>150*250</td>
</tr>
</tbody>
</table>
2.1.1.2. Ball-Wedge Bonding

The thermosonic as well as thermocompression ball bonding method are used for gold wire. Relatively small wires (12 to 75 μm) are used to form a ball (figure 2.2(b)) with a capillary tool for the first bond as shown in diagram 2.4 [20]. The vertical nature of the ball bond on the die reduces the possibility of shorts on the die. But the same tool has to be used for the second bond which results in a rather large circular wedge. Therefore, the substrate bond pads are much larger than the ones for wedge-wedge bonding as shown in table 2.2. Furthermore, due to the needed substrate heating, the thermosonic bonding method is not well suited for laminates due to their rather low glass temperature. The much higher temperature of thermocompression (without ultrasonic energy) is even impossible for most substrate types. On the other hand, the ball bonding process allows any angle for the wires and it is faster, as shown in the comparison in table 2.4. Research of ESEC showed that it is possible to reduce the heating for gold wire bonding down to 50 °C [21]. However, the process window becomes very small and still only a few manufacturers provide it.

![Ball-Wedge bonding process](image)

**Figure 2.4: Ball-Wedge bonding process**

<table>
<thead>
<tr>
<th>Substrate Type</th>
<th>minimum</th>
<th>optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCM-L</td>
<td>150*250</td>
<td>200*300</td>
</tr>
<tr>
<td>MCM-C</td>
<td>150*250</td>
<td>200*300</td>
</tr>
<tr>
<td>MCM-D</td>
<td>120*250</td>
<td>200*300</td>
</tr>
</tbody>
</table>

2.1.1.3. Comparison

The wirebond methods are compared in table 2.3. It is obvious that Al bonding is preferred when it comes to laminates because the needed preheating temperature for Au is higher than most of the materials glass temperatures. When it comes to fine pitch, wedge bonding is preferred but the highest speed and therefore lowest cost is reached with thermosonic ball bonding as shown in table 2.4.
2.1. First Level Interconnects

Table 2.3: Comparison of Wire Bond Methods [20]

<table>
<thead>
<tr>
<th>Method</th>
<th>Wire</th>
<th>Temperature</th>
<th>Pressure</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermocompression</td>
<td>Au</td>
<td>300°C - 400°C</td>
<td>High</td>
<td>Long</td>
</tr>
<tr>
<td>Ultrasonic</td>
<td>Al</td>
<td>Ambient</td>
<td>Low</td>
<td>Short</td>
</tr>
<tr>
<td>Thermosonic</td>
<td>Au</td>
<td>150°C - 225°C</td>
<td>Low</td>
<td>Short</td>
</tr>
</tbody>
</table>

Table 2.4: Comparison of Wire Bond Methods

<table>
<thead>
<tr>
<th>Process</th>
<th>Die Pad Pitch (One Row)</th>
<th>Substrate Pitch on MCM-D (One Row)</th>
<th>Machine Speed [wires/sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ultrasonic Wedge</td>
<td>55 μm</td>
<td>120 μm</td>
<td>5</td>
</tr>
<tr>
<td>Thermosonic Wedge</td>
<td>55 μm</td>
<td>120 μm</td>
<td>3</td>
</tr>
<tr>
<td>Thermocompression Ball</td>
<td>100 μm</td>
<td>250 μm</td>
<td>2</td>
</tr>
<tr>
<td>Thermosonic Ball</td>
<td>60 μm</td>
<td>150 μm</td>
<td>10</td>
</tr>
</tbody>
</table>

2.1.2. TAB

TAB is the acronym of Tape automated Bonding. It connects the die with a substrate or a package using a one or two layer flexible substrate, better known as tape, as shown in figure 2.5. The patterned conductor deposited on a tape is bonded to the corresponding I/O pads on the die. After this inner lead bonding, TAB is connected the substrate in a similar way, called outer lead bonding (OLB).

![Tab Bonded IC](image1.png)  
(a) Tab Bonded IC  
(b) Details of Inner Lead Bonding

Figure 2.5: TAB bonding (Courtesy of IBM)

2.1.2.1. TAB Tape

Many material options are available to adapt a particular TAB tape design to a specific application. Often copper is the preferred conductor material with polyimides as dielectric.
2.1.2.2. Inner Lead Bonding (ILB)

The ILB can either be done with gang bonding (all the leads are bonded simultaneously to the die) or with single point bonding where each lead is bonded to its corresponding die pad. In most cases, ILB requires die bumping (growth of electrochemically deposited bumps on the bonding pads) as shown in figure 2.5(b). Bumping enables and improves bonding operation introducing a higher separation between the tape and the die. New processes will get rid of the need for bumping which is difficult to achieve when dies are not sold in wafer form. Gang bonding is usually carried out by thermocompression (a bond tool heated to 300 - 400 °C applies pression). To achieve uniform bonding with the “Gang” method, a tight temperature control and a very flat tool are required.

Bonding one lead at a time eliminates the temperature uniformity and flatness problems of the gang bonding. For this process thermosonic is commonly used. This process is recommended for dies larger than 10 mm square. Furthermore, single point bonding eliminates custom tooling and allows the repair of missed leads. However, gang bonding is much faster.

2.1.2.3. Encapsulation

After the ILB, the die is encapsulated with epoxies or silicone materials. The encapsulation has to protect the die from chemical and mechanical damage. The encapsulated dies are then delivered on reel or in standard carriers.

2.1.2.4. Test and Burn In

One of the big advantages of TAB is the possibility to test and burn in a die before assembly. It can be done after ILB and encapsulation. Standard test and burn-in sockets are available.

2.1.2.5. Outer Lead Bonding (OLB)

The connection of the TAB to the substrate is called Outer Lead Bonding. In most cases OLB is done by soldering between the leads and the substrate using fine pitch soldering. Other possibilities are thermocompression or thermosonic bonding technique using gang or single point bonding.

2.1.2.6. Conclusion

TAB dies can be tested and burned-in prior to bonding onto the substrate and they can be replaced if repair is needed. Using reeled dies allows a high degree of automation. TAB can be used for high lead-count and fine-pitch VLSI (down to 60 micron). But high lead count introduces a large spreading overhead as the I/Os are usually spread to a pitch of 200 to 300 micron. This pitch can not be improved because more then one row of pads would make the tape too expensive. Furthermore, the design and production of a custom TAB is a big investment. So, when the chip is already available in a TAB package and the bonding area overhead is acceptable or when the improved electrical performance is needed, TAB is a viable alternative.

2.1.3. Flip Chip (FC)

The term flip chip denotes a chip mounting and contacting technology where the ICs are placed face down onto the substrate. To achieve the contact a solderable metal has to be deposited on the chip pads - the bumps. In the assembly process the bumped chips are aligned with the solder lands on the substrate and by a soldering process all bumps are simultaneously connected with the
substrate. Historically, the Flip-Chip process was developed by IBM and used for many years in their System 360 computer. Instead of solder a conductive adhesive can be used and metal deposition can be replaced by a stud formed from a ball bonder. The difference in thermal expansion coefficients between chip and substrate imposes considerable stress on the bumps which leads to fatigue and subsequent fracture after a small number of thermal cycles. Therefore, the gap between the chip and the substrate is underfilled with an epoxy material taking up the stress from thermal cycles and increasing the reliability of Flip Chip contacts by several orders of magnitude. A cross section of a Flip Chip is shown in figure 2.6.

![Flip Chip Cross Section](image)

**Figure 2.6: Flip chip Bonding**

Special care must be taken for ICs with higher power dissipation. The usual thermal path is through the bumps into the substrate. To lower the thermal resistance, additional thermal bumps can be placed and/or a heat sink can be mounted on the back side of the ICs.

2.1.3.1. Summary

Flip chip allows the highest packaging density among all the assembly technologies because there is no overhead to the IC size.

The conductor lines on the substrate must not have a larger pitch than the chip pads (typically 150 - 200 micron) and require a planarity within a few microns over the size of the chip to achieve high yields in the assembly process. These requirements impose quite severe restrictions on the choice of microvia substrates suitable for FC mounting. They are not fulfilled by standard FR-4 substrates used for PCBs.

The low inductivity of FC joints compared to wire bonding in table 2.5 results in excellent electrical properties and makes FC suitable for high frequency applications and submicron CMOS technologies with high transient currents in the power/ground system.

However, because there is no interposer between IC and substrate, underfill is needed to relief the induced thermal stress.

2.1.4. Chip Size Package

Chip size packages (CSP) are not a first level interconnect but miniaturized packages. They were developed out of the now famous ball grid arrays (BGA) and have the area pins in common with them. CSPs have a ball pitch smaller than 1 mm and the package should be smaller than the die area plus 10%. The CSPs can either be built with an interposer, or the IC is rerouted from peripheral to area pads. As shown on the left side and in the center of figure 2.7, the die can be bonded with flip chip or TAB onto the interposer. Wire bond is also possible but enlarges the interposer. A rerouted IC is shown on the right side of figure 2.7.

As CSPs are packages they can be tested and a die shrink often does not change the package. On the other hand they are higher than bare dies as shown in figure 2.7 and they add an additional
package layer. This layer might degrade the speed, as often peripheral pads are routed to the inside to form an area connection and have than to break out on the board again. CSPs are well suited for memory ICs because all the balls can be placed on the die with a moderate pitch. It even could become a standard for them. However, up to now the CSPs are about 10% more expensive than standard packages such as TS(S)OP.

![Figure 2.7: CSP Types](image)

### 2.1.5. Comparison of First Level Interconnects

Compared to other connection techniques wirebonding is suitable for chips with peripheral pads when the minimum pitch and electrical performance is sufficient. TAB reaches slightly smaller pitches and can exhibit less inductance and capacitance than wirebonds. But the NRE cost for TAB are high and for multiple rows on the die it is even worse. Furthermore, the footprint is generally larger than for wirebonding. But TAB supports pretest of dies.

Best electrical performance is available with Flip Chip bonding as compared in table 2.5. FC further provides smallest size and highest I/O density. But it is difficult to use it with fine pitch ICs and the infrastructure is not very well established.

The availability of a mature infrastructure for wirebonding as well as relatively low cost and ease of design makes wirebonding a viable option for the first level interconnect as long as CSP do not become standardized and are not available at lower cost.

![Table 2.5: Electrical Parameters of Bonding Technologies](image)

<table>
<thead>
<tr>
<th>Technology</th>
<th>Inductance [nH]</th>
<th>Capacitance [pF]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flip Chip</td>
<td>0.1 - 0.2</td>
<td>0.02 - 0.03</td>
</tr>
<tr>
<td>CSP with interposer</td>
<td>0.3 - 2.3</td>
<td>0.05 - 0.34</td>
</tr>
<tr>
<td>CSP rerouting</td>
<td>&lt; 0.01</td>
<td>&lt; 1</td>
</tr>
<tr>
<td>Wire bond</td>
<td>1 - 5</td>
<td>0.2 - 0.6</td>
</tr>
<tr>
<td>TAB</td>
<td>1.3</td>
<td>0.2 - 0.6</td>
</tr>
<tr>
<td>QFP</td>
<td>6 - 7</td>
<td>0.5 - 1</td>
</tr>
<tr>
<td>PGA</td>
<td>7.7 - 15</td>
<td>1 - 3</td>
</tr>
</tbody>
</table>

### 2.2. Substrate Technologies

The first level interconnects presented in the section before require a higher density interconnect than the well known printed circuit board. The required interconnect technologies are called substrates and they can be divided into three main categories: laminate, ceramic and thin film. The first two are named after their dielectric material used. Thin film on the other hand describes the
production process. However, the technologies can be mixed and laminate as well as ceramic are categories with a lot of members. All technologies can be characterized by the parameters shown in figure 2.8.

![Design Rules Diagram](image)

**Figure 2.8: Substrate Parameters for HDP Substrates**

### 2.2.1. Laminate

MCM-L stands for laminated substrates. The technology is essentially a high-end or improved PCB technology which satisfies the MCM requirements in terms of interconnection density and cost. The main obstacle to improve the wiring density is the via land size as shown in figure 2.9. In table 2.6 typical design rules for this technology can be found.

![Cross Section and Top View Diagram](image)

**Figure 2.9: PCB Substrate**

### 2.2.2. Sequential Build-Up

The traditional PCB technology is massively improved by build-up layers. These Sequential Build Up (SBU) substrates use microvias drilled by means of laser ablation, plasma etching or photodefinable dielectric. SBU layers not only provide very small vias but also improve the design rules as
the microvias do not need a thick metalization. Furthermore the vias often cross only one dielectric layer and therefore do not block all layers below.

Figure 2.10 shows the cross section of an SBU substrate. The core is a standard PCB with or without drilled holes. Onto this core additional dielectric layers are added symmetrically on both sides by laminating foils or adding a liquid dielectric. Then vias are opened in the dielectric and metalized, the metal layer is patterned and the steps are repeated for another SBU layer if needed. SBU is not only one type but there exists a bunch of customizable possibilities. The following parameters characterize them:

- Number of SBU layers
- Designrules: core, top layer core, SBU, top layer SBU
- Mechanically drilled vias: in core, through whole substrate, both
- Dielectric: liquid or foil
- Microvias: Laser, Plasma etching, Photovias

Typical design rules can be found in table 2.7. The selection of the through hole via type has to be done carefully as it can define the substrate cost. The mechanically drilled vias need thick metalization layers to get a reliable hole metalization. As thick metal layers do not allow fine structures (underetching is up to metal thickness) through hole vias through the whole substrate should be avoided if fine structures on the top SBU layer are required. However, laser vias are expensive if numerous. They should not often be used to connect to the drilled vias in the core just to cross to the opposite substrate side. Using both options – drilled via in the core as well as through the whole substrate – is even worse because the substrate has to be mechanically drilled twice.

2.2.3. Thin Film (MCM-D)

Thin film substrates are no longer named after their dielectric or base material but after their manufacturing process. Thin film materials such as dielectric or metal are deposited and processed on a dimensionally stable base such as silicon, aluminium, alumina, aluminium nitride or, in the near future, laminate. With its very thin metal and dielectric layers it is the technology with the highest routing density per layer as summarized in table 2.8. Due to the deposition process, this substrate...
technology is also known as MCM-D. The processes used are common in semiconductor industry. For the metal layers such as power planes or signal layers an adhesion layer followed by a seed layer is sputtered or vacuum evaporated. Then either electroplating or sputtering is used to selectively grow conductor thickness. Onto these copper or aluminium layers, dielectric material such as silicon dioxide or polymers such as polyimide or BCB is used to separate the metal layers. Whereas the dioxide is grown, the polymers are spin coated and then baked. The polymers provide a low relative dielectric constant, thus having the advantage that conductors have adequate impedances in spite of the very thin dielectric layers. The vias are then formed in a photolithography process. Integrated resistors and capacitors are possible with additional layers as described in section 2.3. If silicon is used as base, active circuits such as I/O protection, memory or slow logic can be integrated into the substrate. This chip-on-chip technology is very dense but expensive. Another option is to build the interconnect layers onto ICs and mount some more on the top. This option is also very expensive because bad ICs or substrate layers reduce the yield of the already expensive solution.

### 2.2.4. Hybrid

A Hybrid is not a real HDP technology, but it can be regarded as its predecessor because bare die attach on hybrids has a long tradition. On a fired ceramic base, a small number of layers is deposited by screen printing of a conductive paste. After the printing of the first layer it is fired and then a glass layer is screen printed as isolation layer and fired, too. The designrules can be found in table 2.9. This technology is famous for its printed and trimmed precision resistors from 1Ω to 10 MΩ as described in section 2.3.
Chapter 2: Technologies

2.2.5. Cofired Ceramic

Cofired ceramics are made from unfired (green) ceramic. The dielectric layers are punched ceramic sheets. The lines are then screen printed onto this layers and the vias filled. Finally, all layers are stacked and simultaneously fired under pressure. The firing temperature depends on the printing paste used. For high temperature cofired ceramics (HTCC) gold pastes are used. These pastes and the ceramic used require a higher temperature (1800°C) than copper pastes (860°C) used for low temperature cofired ceramics (LTCC). A crosssection is shown in figure 2.13(a) and typical design rules can be found in table 2.10.

---

**Figure 2.11: Cross section Thin Film Substrate (MCM-D)**

**Table 2.8: Thin Film Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Width</td>
<td>10 to 40 μm</td>
</tr>
<tr>
<td>Line Space</td>
<td>20 to 60 μm</td>
</tr>
<tr>
<td>Via Land Diameter</td>
<td>30 to 80 μm</td>
</tr>
<tr>
<td>Metall Thickness</td>
<td>2 to 5 μm</td>
</tr>
<tr>
<td>Dielectric Thickness</td>
<td>2 to 15 μm</td>
</tr>
<tr>
<td>Dielectric Constant $\varepsilon_r$</td>
<td>2.7-3.5</td>
</tr>
<tr>
<td>Number of layers</td>
<td>2 to 5</td>
</tr>
<tr>
<td>Thermal Conduction</td>
<td>high</td>
</tr>
<tr>
<td>Substrate Cost</td>
<td>high</td>
</tr>
</tbody>
</table>

---
### Table 2.9: Design rules for Thick Film Hybrid (Saini 1999)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Width/Pitch</td>
<td>&gt; 125 / 250 μm</td>
</tr>
<tr>
<td>Via Diameter</td>
<td>&gt; 100 μm</td>
</tr>
<tr>
<td>Via Pad Diameter</td>
<td>&gt; 200 μm</td>
</tr>
<tr>
<td>Layer Count</td>
<td>1 to 3</td>
</tr>
<tr>
<td>Dielectric Thickness</td>
<td>40 μm</td>
</tr>
<tr>
<td>Metal Thickness</td>
<td>8 to 10 μm</td>
</tr>
<tr>
<td>Dielectric Constant εₚ</td>
<td>6 to 10</td>
</tr>
<tr>
<td>CTE</td>
<td>7.9 to 10 ppm</td>
</tr>
<tr>
<td>Thermal Conduction</td>
<td>high</td>
</tr>
<tr>
<td>Substrate Cost</td>
<td>moderate</td>
</tr>
</tbody>
</table>

### Table 2.10: Design rules for Cofired Ceramics (Saini 1999)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line Width/Pitch</td>
<td>&gt; 125 / 250 μm</td>
</tr>
<tr>
<td>Via Diameter</td>
<td>&gt; 120 μm</td>
</tr>
<tr>
<td>Via Pad Diameter</td>
<td>&gt; 200 μm</td>
</tr>
<tr>
<td>Layer Count</td>
<td>15 to 30</td>
</tr>
<tr>
<td>Dielectric Thickness</td>
<td>40 to 100 μm</td>
</tr>
<tr>
<td>Metal Thickness</td>
<td>8 μm</td>
</tr>
<tr>
<td>Dielectric Constant εₚ</td>
<td>6 to 10</td>
</tr>
<tr>
<td>CTE</td>
<td>7.9 to 10 ppm</td>
</tr>
<tr>
<td>Thermal Conduction</td>
<td>high</td>
</tr>
<tr>
<td>Substrate Cost</td>
<td>high</td>
</tr>
</tbody>
</table>
2.2.6. Ceramic Photodefinable

Recently, new thick film materials were developed that are patterned by photolithographic instead of using screenprinting[23][24]. Photodefinable ceramics have finer line pitches than other ceramic technologies as summarized in table 2.11.

In photodefinable technologies, the conductor paste is printed in a blank pattern onto a ceramic substrate and dried. Then the layer is patterned by UV light and finally fired. The firing process shrinks the line width as shown in table 2.11. The dielectric paste is also printed in a blank, unpatterned layer over the previously processed conductors. After levelling the printed layer is dried. The via formation is done using a Diffusion Patterning process. Here via dots are printed of the patterned paste which is then diffused into the dielectric (condition 20 minutes at 80 degree C). The developed dielectric layer under the dot can then be washed away and is fired. If the resulting thickness is not enough, the dielectric process can be repeated. Finally the vias are filled with paste and fired. For additional layers the process is repeated.

Table 2.11: Designates for Photodefinable Ceramics (Summa 1999)

<table>
<thead>
<tr>
<th>Line Width/Pitch</th>
<th>&gt; 57/75 on mask)/120 to 92/115 on mask)/200 (\mu m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Via Diameter</td>
<td>&gt; 200 (designed as 150 (\mu m))</td>
</tr>
<tr>
<td>Via Pad Diameter</td>
<td>&gt; 250 (\mu m)</td>
</tr>
<tr>
<td>Layer Count</td>
<td>2 to 5</td>
</tr>
<tr>
<td>Dielectric Thickness</td>
<td>15(one print) to 36(printed twice)(\mu m)</td>
</tr>
<tr>
<td>Metal Thickness</td>
<td>7 to 9 (\mu m)</td>
</tr>
<tr>
<td>Dielectric Constant (\epsilon_r)</td>
<td>6 to 10</td>
</tr>
<tr>
<td>CTE</td>
<td>8 to 10 ppm</td>
</tr>
<tr>
<td>Thermal Conductivity</td>
<td>high</td>
</tr>
<tr>
<td>Substrate Cost</td>
<td>moderate</td>
</tr>
</tbody>
</table>
2.2.7. Comparison of Substrate Technologies

This overview showed that laminate substrates are by far the lowest cost substrates. However, they often do not fulfil all the requirements such as thermal conductivity or wiring density. Figure 2.14 shows a comparison of the three main substrate types laminate, ceramic and thin film. It can be seen that thin film is by far the most dense technology but is limited in the layer count. SBU technologies also allow quite fine lines but their density is limited by the via land diameter and the number of layers. Ceramic substrates have smaller vias but much larger line pitches which can be compensated by the number of layers. However, ceramic technology is much more expensive than SBU laminates.

![Figure 2.14: Comparison of Laminate, Ceramic and Thin Film Substrates](image)

A more detailed comparison is shown in table 2.12. It shows that thin film is more expensive than ceramic. However, if thin film allows to build a system on a very small size, the substrate cost become low.

<table>
<thead>
<tr>
<th>Dielectric</th>
<th>Thin Film</th>
<th>HTCC</th>
<th>LTCC</th>
<th>Laminate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metalisation</td>
<td>Polymid, BCB, Hf$_2$O$_5$</td>
<td>Alumina Paste</td>
<td>Alumina</td>
<td>Epoxy</td>
</tr>
<tr>
<td>typ. (max) Layers</td>
<td>3(5)</td>
<td>30(up to 50)</td>
<td>15(up to 30)</td>
<td>8(up to 30)</td>
</tr>
<tr>
<td>Line pitch [μm]</td>
<td>50</td>
<td>200</td>
<td>200</td>
<td>180</td>
</tr>
<tr>
<td>Time of flight [ps/cm]</td>
<td>62</td>
<td>102</td>
<td>90</td>
<td>72</td>
</tr>
<tr>
<td>Line resistance [Ω/cm]</td>
<td>1.3 - 3.5</td>
<td>0.8</td>
<td>0.2</td>
<td>0.06</td>
</tr>
<tr>
<td>Cost</td>
<td>high</td>
<td>moderate</td>
<td>moderate</td>
<td>low</td>
</tr>
</tbody>
</table>

2.3. Integrated Passives

Passive components such as resistors, capacitors, inductors, filters and arrays or combinations of them are either assembled in discrete form on the HDP substrate or can be integrated into it. The decision of which strategy to use depends on the number and the value of the passive components,
the availability and consumption of the substrate area as well as the stability, accuracy and Q-factor required. Case studies can be found in [25]. The following subsections provide a short overview of passive components for thin- and thick-film as well as laminate technology. These technologies enable also complete analog networks, couplers and antennas [26].

2.3.1. Integrated Resistors

Thin film techniques are very well suited for the integration of resistors[27]. They satisfy the miniaturisation requirements and have an excellent matching if designed with care. They are long term stable and have a high reliability as summarized in table 2.13 and 2.14. Integrated resistors are built by depositing a resistive layer and by patterning it as shown in figure 2.15. On both ends the resistive lines are contacted by metal.

Thick film resistors have high temperature coefficients and provide more noise than thin film resistors but they provide a much larger range of values. For them, a resistive paste is printed and then fired [28].

On laminate substrates a polymer paste can also be used to print resistors [29]. However, the accuracy is only ±25%. Thus it is only suited for pull-up/down resistors. Furthermore, this paste is not stable over lifetime and it is sensible to humidity. Another technique used for laminates is the lamination of a one micron thick NiP foil which is patterned afterwards [30]. These foils have a sheet resistance of 100 to 100kΩ per square. PCB base material with a resistive layer is available under the trade name OHMEGA PLY[31][32].

<table>
<thead>
<tr>
<th>Type</th>
<th>Resistive Materials</th>
<th>Resistor Range</th>
<th>Temp. coeff.</th>
<th>Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thin film</td>
<td>SiC, SiC, TaN</td>
<td>10Ω-10MΩ</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Thick film</td>
<td>CuO₂, BaRuO₃, Bi₂Ru₂O₇</td>
<td>1Ω-1GΩ</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Laminate Printed</td>
<td>Polymer with Carbon Particles</td>
<td>20Ω-150kΩ</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Laminate Foil</td>
<td>NiP foil</td>
<td>10Ω-1GΩ</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

2.3.2. Integrated Capacitors

Integrated capacitors can either be built as plate capacitor on two layers (figure 2.16(a)) or as interdigitated structure (figure 2.16(b)) on one or two layers.

<table>
<thead>
<tr>
<th>Type</th>
<th>matching</th>
<th>absolute</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>trim.</td>
<td>not trim.</td>
</tr>
<tr>
<td>Thin film</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Thick film</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Laminate Printed</td>
<td>n/a</td>
<td>-</td>
</tr>
<tr>
<td>Laminate Foil</td>
<td>n/a</td>
<td>-</td>
</tr>
</tbody>
</table>
Metal-Insulator-Metal (MIM) are either built between the standard metal layers using the same dielectric as for wire isolation, or a special dielectric layer is applied featuring a very high $\varepsilon_r$. The latter technique allows capacitor densities of more than 50 nF/cm$^2$ in thin film technology as shown in Table 2.15. For thick film substrates ferroelectric materials are used to build capacitors up to 100 pF (absolute). With new polymer-ceramic composites capacitor densities of more than 22 nF/cm$^2$ are possible on laminates [33].

Interdigitated capacitors [34][35] have proven to be useful components on high $\varepsilon_r$ base materials for microwave circuits due to their simplicity of construction, relatively high Q and good repeatability. The capacitor is formed by the fringing field of an interdigital gap between fingers as shown in Figure 2.16(b). Because of size limitations, the capacitance values obtained are typically less than 1 pF. Design formulas can be found in [34][35][36].

Table 2.15: Integrated MIM Capacitors

<table>
<thead>
<tr>
<th>Type</th>
<th>Dielectric Materials</th>
<th>Range</th>
<th>Tolerance</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thin film</td>
<td>SiO2, SiN, Al2O3, BaTiOy</td>
<td>5 nF - 50 nF/cm$^2$</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>Thick film</td>
<td>Ferroelectric materials</td>
<td>0.2 - 100 pF (absolute)</td>
<td>-</td>
<td>- -</td>
</tr>
<tr>
<td>Laminate</td>
<td>Polymer with inorganic filler</td>
<td>1 - 22 nF/cm$^2$</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Interdigitated</td>
<td>Si, Al2O3</td>
<td>&lt; 1 pF (absolute)</td>
<td>++</td>
<td>++</td>
</tr>
</tbody>
</table>

Figure 2.15: Layout of Thin Film Resistor

Figure 2.16: Integrated Capacitors (Courtesy of Smiths Conventum)
2.3.3. Integrated Inductors

Embedded inductors are usually made with copper lines using a spiral design on one layer and a connection to the center on an inner layer as shown in figure 2.17. On ceramic and polyimide base materials, Q values from 20 to 90 at 1 GHz were reported [37]. However, for thin film on silicon a big effort is required to design inductors featuring quality factors above 20 [38]. In all technologies inductor values up to 100 nH are reached. However, it is unlikely that inductors are far away from other circuits. Thus, special consideration must be given to the effects of other layers of insulating and conductive materials. Because of the inductor’s spiral construction, the magnetic fields will extend far past the device boundaries. Any conductive material that is in the vicinity of the inductor and through which the magnetic field passes will distort the inherent magnetic field and affect the inductance. Empiric design equations can be found in [39], [40] and [41]. However, these equations are complicated and still not very accurate[42][41]. A careful design requires numeric simulations especially when more than one dielectric material is in the cross section.

<table>
<thead>
<tr>
<th>Type</th>
<th>Range</th>
<th>Noise</th>
<th>Tolerance</th>
<th>Q-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thin film</td>
<td>0 - 100 nH</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Thick film</td>
<td>0 - 100 nH</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

(a) Top View  
(Courtesy of Summit  
Consortium)

(b) Cross Section  
(Courtesy of IMEC)

*Figure 2.17: Thin Film Inductor*

2.4. Summary

In this chapter the available bonding and substrate technologies were presented. Besides interconnecting active components and passives on a very small area, these technologies also allow to integrate the latter into them. However, this overview did not yet answer the question which technology should be used for a system. This was not the aim of this chapter because there are so many options as illustrated in figure 3.1 and the suitable technologies depend on the application specific requirements.

To perform a system partitioning and to select the optimum technologies, a design space exploration is performed as described in the next chapter. The design space exploration first sets up
2.4. Summary

a tree of suitable implementations and extracts parameters that helps selecting one of them for implementation.
A design space exploration is required in the high level design for high density packaging. High level design is the most important part of a design cycle where, prior to detailed simulation activities and layout, the concepts and the build-up of a system are defined. The high level design starts with specification of the system features, continues with system architecture design, followed by hardware and software specifications and ends with the development of netlists and other information needed to begin physical design. It is the most important part because 80% of a product’s final cost is determined by decisions taken in the first 10% of the design cycle time [43].

The goal of designers is to find a design which balances size and performance with ease of manufacturing to minimize the cost. This becomes more and more difficult because numerous component and packaging alternatives, as shown in figure 3.1, have to be considered. A straightforward approach (figure 3.2) as it is widely used in industry is no longer possible because interconnection and packaging technology define the performance and therefore have to be taken into account. So, physical aspects have to be brought into high level design. Furthermore, it is not very obvious from the beginning which combination of partitioning and packaging alternatives lead to an optimal solution. Its selection is done in a feasibility analysis. This analysis performs a design space exploration and is integrated into the design flow as described in the section below.

Most aspects are illustrated with the design process of an Pentium MCM for embedded computing [44].

### 3.1. Design Flow

A design flow for advanced packaging should include a feasibility study. Feasibility analyzes are used to develop a system specification into a product. As it is not obvious from the beginning which technologies the system should be built with, several alternatives have to be generated and analyzed. This design space exploration is integrated in a design flow consisting of the following steps:

1. Determine the system requirements and goals.
2. Determine the suitable alternatives for packaging, partitioning and floorplaning (Set up a tree of feasible alternatives).
3. Evaluate the performance and cost of each alternative.
4. Make the final decision where only ONE alternative should be chosen.
To do so, the design flow has to be adapted in two ways: It must become iterative in an early stage to refine all the analysis done and the decisions taken. Furthermore, the idea of taking decision after decision to end up with one possible solution as shown in figure 3.2 has to be abandoned, because it is not obvious which combination is the optimal solutions. Instead, a tree of all possible alternatives is created as shown in figure 3.3. In each step some branches are discontinued when they are obviously infeasible.
In order to analyze different partitioning options, to check the possible packaging strategies and to prepare the cost estimation, a detailed feasibility analysis is performed as described in the remainder of this chapter.

3.2. Procedure

Starting from a specification, the required functionality has to be partitioned into components such as standard components, ASICs or programmable logic. Often there are several alternative (sets) that fulfil the requirements.

These sets are the basis to partition the components of a set into subsystems further referred to as partitions.

As shown in figure 3.3 different build-up technologies are generated and evaluated for the partitions as it is not obvious which alternative best fulfills the specifications.

If there are alternatives for a set that fulfil the specifications they are enhanced for testability.

When all the technical aspects were considered for an alternative and the specifications are more or less fulfilled, the cost can be calculated based on the extracted parameters.

In a cost-performance trade-off one or several solutions are finally selected for further consideration.

After this decision, the layout can be started. Furthermore, the proposed methodology ensures that the design rules and the cross section\(^1\) of a substrate are known. Therefore, a layout has to be done only once. To do so, the alternatives have to be generated and analyzed by considering the following key attributes:

1. Bonding Option
2. Substrate Types
3. Component Limited Size
4. Power Distribution
5. Thermal Analysis
6. Routing Analysis
7. Signal Distribution, FMI and FMC
8. Design for Testability (DFT)
9. Interface to the next level (Package, Direct Soldering...)
10. Yield and Cost

\(^1\)Layer stackup
Several implementations have to be created and compared iteratively by extracting and weighting these attributes. Therefore, the study has to be done several times to create and reject more and more options and to decide and analyse more and more details using preciser and complexer models in each pass. The first models are heuristic and not all of the above attributes are analyzed. In a second pass more and analytic models are used and for some details even simulations are performed. Most aspects are illustrated with data from the Pentium MCM described in the next section 3.3.

3.3. The Pentium MCM

A typical industrial PC based on a “Pentium” CPU is illustrated in the block diagram in figure 3.4. The core of the system is the processor. It is supported by the chip set consisting of the system controller $MTSC$, the data path $MTDP$ and the PCI to ISA bridge $MPIIX$ in this example. In addition to the main memory (DRAM) these systems include SRAM tightly coupled to the processor, the so called second level cache. The component data can be found in table 3.1.

The components are interconnected with three main busses, namely

1. Host Bus: This is the high-speed bus interconnecting the cache, the two data path managers ($MTDP$) and the system controller ($MTSC$).

2. PCI Bus: Standard Peripheral Interconnect Bus

3. Extended I/O Bus: a reduced ISA Bus. This bus has been developed out of the well known AT-Bus and enables slow peripheral equipment to communicate with the system.

Two controllers bridge these busses: the system controller ($MTSC$) and the PCI-to-ISA XCELERATOR ($MPIIX$), the so called PCI to ISA Bridge.
3.3. The Pentium MCM

### Table 3.1: Pentium MCM Components

<table>
<thead>
<tr>
<th>Component</th>
<th>$C_{Length}$ [μm]</th>
<th>$C_{Width}$ [μm]</th>
<th>$N_{I/O}$</th>
<th>$C_{PP}$ [μm]</th>
<th>Power dissipation [W]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pentium</td>
<td>9900</td>
<td>9100</td>
<td>173</td>
<td>75</td>
<td>7.5</td>
</tr>
<tr>
<td>TSC</td>
<td>6000</td>
<td>6000</td>
<td>179</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>2*TDP</td>
<td>3680</td>
<td>3400</td>
<td>80</td>
<td>120</td>
<td>1</td>
</tr>
<tr>
<td>4*PBSRAM</td>
<td>7700</td>
<td>6900</td>
<td>72</td>
<td>200</td>
<td>1</td>
</tr>
<tr>
<td>ASRAM</td>
<td>5350</td>
<td>3170</td>
<td>26</td>
<td>200</td>
<td>0.5</td>
</tr>
<tr>
<td>Substrate</td>
<td>32000</td>
<td>32000</td>
<td>189</td>
<td>300</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td>1015</td>
<td></td>
<td>16.5</td>
</tr>
</tbody>
</table>

#### 3.3.1. Design Goals

The aim of the Pentium MCM is to build a very small building block for embedded computing. Thus, the requirements are as follows:

1. Modular building block suitable for as many applications as possible
2. Small size and height
3. Encapsulating the complexity and relaxing requirements for motherboards
4. Moderate cost
Whereas number one depends on the partitioning described in the next section, the others also depend on the selected technologies.

3.4. Partitioning

Partitioning of functionality into chips is a fundamental design concern for high performance systems. The challenge is to determine the optimum number of chips for a given functionality, based on a cost/performance trade-off. Several component sets are created by partitioning the system into existing components including microcontroller, programmable logic (PLDs, FPGAs) and new ASICs.

When it comes to ASICs, even more aspects have to be considered. Putting a large amount of functionality into a single chip may provide electrical performance and system size advantages. But this strategy often results in large dies with low yields and therefore high cost. Realizing the same functionality using a large number of small dies leads to a larger system size and possible performance degradation due to larger interconnection delays.

System attributes do not depend on component selection only. Packaging strategies, including chip attach methods and substrate type, also influence size, speed and cost. To evaluate these attributes, the component sets are first assigned to functional building blocks and then these functional blocks are assigned to Multi Chip Modules or directly to a PCB which can be reused in other systems. The modules and the PCBs themselves are further assigned to a PCB or a backplane.

After the partitioning task described in the remainder of this section, the optimal physical implementation will be analyzed in the feasibility study starting in section 3.5.

3.4.1. Partitioning Functionality into Chips

The first step in partitioning a system is to assign functionality to silicon. By using a component library a functional system description (for example in behavioural VHDL) digital systems could automatically be translated into a component list, by dividing the design into existing MCMs, RAMs, CPUs, decoders, controllers, etc. and logic components for the remaining glue logic. But mostly, this does not lead to a good solution. Glue logic may be implemented in a programmable logic device or even as a part of one or several ASICs. The same aspects have to be considered for other system parts such as CPUs, which can be implemented as single components, cores in an ASIC or built using several chips. This has been the traditional method for mainframe computers.

The aim of this first step is to create several component lists (sets) which could possibly lead to a successful implementation. So, the component space is set up. Even uncommon component combinations can be used because advanced packaging opens up new partitioning options. Only some combinations can already be skipped at this stage due to chip size, timing constraints or cost, because many attributes heavily depend on packaging technology.

3.4.1.1. Programmable Logic

Programmable logic devices are either FPGAs or PLDs. PLDs are more common for controllers than FPGAs and FPGAs suit better to data path architectures and are an alternative to ASICs for small series. The wanted device could be selected by comparing the required number of pins, number of flip-flops and number of logic blocks to available ones. However, a successful implementation often fails because of details such as the number of output enables. Therefore, it is safer to feed the description into synthesis tools - which often have an interface to VHDL - to get a list
of possible devices. The system parts intended to be implemented in programmable logic could be partitioned themselves into several devices instead of using one very big and expensive component.

3.4.1.2. ASICs

The goal is to find the optimum number of chips by partitioning the functional description into several dies. Thus, the area of each partitioned die will decrease, resulting in an improved yield for each one. On the other hand, the off chip interconnect delay as well as the system size are often larger than in a single chip solution. Furthermore, they both depend on the interconnect substrate type and the chip attachment. Nevertheless, several chips in a MCM can be less expensive than a single large one due to improved die yield. To model the physical implementation, the chip sizes and their yields have to be estimated if design data is not accurate enough.

**Die Size Estimation** There are two possibilities to estimate the die size if no detailed specification is available: Rent’s rule or Sum of Areas.

The easy approach is ‘sum of areas’[45]. Starting from a functional description, the known sizes of building blocks such as adders, multipliers, multiplexers are summed up to get the core area. By estimating the number of I/Os and the number of power pins or the signal to ground ratio, the die area can be determined using equations 3.1-3.3.

\[
\text{peripheral area I/O limited} = \left( 2CP_{Length} + CP_{PP} \left[ \frac{CN}{4} \right] \right)^2 \tag{3.1}
\]

\[
\text{peripheral area core limited} = \left( 2CP_{Length} + \sqrt{ACore} \right)^2 \tag{3.2}
\]

\[
\text{array area I/O limited} = \left( \sqrt{CN} + 1 \right) \times CP_{PP} \tag{3.3}
\]

\[C_{Length} = \text{length of a peripheral bond pad on the die}\]
\[C_{PP} = \text{minimum center-to-center pitch of bond pads on the die}\]
\[C_N = \text{total number of die I/O}\]
\[A_{core} = \text{core area of an IC (without I/O buffers and pads)}\]

Another approach was developed by E.F. Rent of IBM using an empirical relation between the number of pins and the number of gates in a circuit[46]:

\[
C_N = K_pN_g^\beta \tag{3.4}
\]

where \(C_N\) is the number of pins required for a circuit with \(N_g\) gates. The constants \(K_p\) and \(\beta\) are empirically derived and depend on the architecture, organization and implementation of the design: Furthermore, the core area \(A_{core}\) is estimated from the number of gates \(N_g\) and the effective gate dimensions \(d_g\) in (3.5).

\[
A_{core} = d_g^2N_g \tag{3.5}
\]

Finally, the die size is calculated using above equations 3.1-3.3.

**Yield Estimation** The die yield \(Y\) can be estimated by the generalized negative binomial distribution[47] (3.6). However, this equation is not valid if redundancy is used to increase initial die yield. Redundancy is often implemented as redundant rows/columns on a RAM. The clustering parameter \(s\) measures how correlated the failures are as expressed in equation 3.7 [48]. The value of \(s\) ranges from 0.2 to 3.4.
### Chapter 3: Design Space Exploration

<table>
<thead>
<tr>
<th>System or Chip Type</th>
<th>Exponent ( \beta )</th>
<th>Multiplier ( K_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static memory</td>
<td>0.12</td>
<td>6</td>
</tr>
<tr>
<td>Micro processor</td>
<td>0.45</td>
<td>0.82</td>
</tr>
<tr>
<td>Gate array</td>
<td>0.50</td>
<td>1.9</td>
</tr>
<tr>
<td>High-speed computer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chip and module level</td>
<td>0.63</td>
<td>1.4</td>
</tr>
<tr>
<td>Board and system level</td>
<td>0.25</td>
<td>82</td>
</tr>
</tbody>
</table>

*Table 3.2: Rent’s Constants*

\[
\begin{align*}
Y & = \frac{1}{(1 + sDA)^{1/s}} \quad (3.6) \\
s & = \text{clustering parameter} \\
D & = \text{average defect density} \\
A & = \text{area of the die}
\end{align*}
\]

#### 3.4.2. Component Set Partitioning and Assigning Partitions to Modules

Before assigning component sets to modules, they have to be grouped in partitions which can then be assigned to modules or PCBs. The goal of this functional partitioning task is to create rather small function blocks with a small interface between each other. Therefore, partitioning should be done considering the following aspects:

- A partition is a subsystem which could be used stand-alone.
- It should have a reasonable size and a small interface to each other.
- The partitions should be reusable as multifunctional building blocks in other designs.
- Critical paths as well as accurate analog circuits should not leave any module because inter-partition interconnections inevitably lead to longer lines and, hence, longer delays as well as signal distortions from impedance mismatches.
- When the partitions are assigned to modules or PCBs, the availability of components in the required packaging form (like bare dies for MCMs) is another major concern.

Based on the above principles the Pentium MCM is partitioned in the next subsection.

#### 3.4.3. Partitioning of the Pentium MCM

As described in section 3.3, the Host-Bus incorporates all high speed interconnections (see also figure 3.4). Therefore, the following blocks constitute a partition which conforms to all the above requirements:

- Pentium processor
- Second level cache: PBSRAM (Cache) and SRAM (Tag)
3.5. Bonding Options

- System controller: MTSC
- Data Path controller: MTDP

Moreover, this partition provides a common interface to the main memory and the PCI bus. The minimization of I/O count would lead to an integration of the main memory into this partition. However, this is not advisable as re-usability and upgradability of such a partition (i.e. the Pentium system plus the main memory) would be limited. Instead, the main memory is seen as a separate partition. This means that the Pentium partition can be easily combined with a number of different instances of the main memory partition (each with a different memory depth), resulting in different overall system configurations, which also ensures easy upgradability. This partition was therefore selected for the realisation of this system.

The partitions are now assigned to modules (i.e. MCMs or PCBs). The "Pentium" chip with only four percent silicon has the biggest packaging overhead in a single chip package. To get a partition of reasonable size, the partition consisting of the Pentium, its chip-set and the cache should be assigned to a MCM. However, one of the major concerns when assigning a (hypothetical) partition to a MCM is the availability of ALL dies as unpackaged components. In this specific design, all Intel components were available as so-called "Smart Dies" (i.e. dies which conform to the same specifications as their packaged counterparts) and all SRAMs were available as bare dies from Micron (synchronous) and Samsung (asynchronous).

To check the above partitions and to finalise this decision and to decide the implementation, a feasibility analysis is done in the remaining sections of this chapter.

3.5. Bonding Options

A detailed description of the performances of the different bonding options can be found in chapter 2.1. The aim in this section is to model the size as well as other impacts of the bonding method to the system design. The effective component sizes \( A_{Esc} \) depend on the selected bonding method and to a minor extent on the substrate selection. In equation (3.8) is \( A_{Pads} \) the component size and \( O_{Esc} \) the overhead due to bonding and the escaping that enlarges the component size as shown in figure 3.5.

\[
A_{Esc} = \left( \sqrt{A_{Pads}} + O_{Esc} \right)^2
\]  

(3.8)

Due to the fact that the substrate selection depends on the effective component size, the bonding method has to be chosen first. After a first substrate selection the calculation of the effective component size can be refined or more detailed models as described in chapter 4 can be used. Furthermore, it is very important to choose whenever possible only compatible attachment methods for all chips to ensure good manufacturability.

3.5.1. Wire Bond

Wire Bonding is the most mature assembly method and suits to ICs with a peripheral pad configuration. Nevertheless, it is a sequential process and it adds a slight overhead to the IC area. As rule of thumb for wire bonding the die sizes have to be increased on each length about by 1.8 mm for one bond row and 3 mm for two bond rows. Generally, simple chips need one row and complex chips featuring more than 60 pads need two or even three rows if using low density substrates.

The number of rows can also be calculated with equation (3.9) using bond pad pitch (\( BP_{PP} \) and
die pad pitch ($C_{PP}$). For orthogonal wire bonding, the overhead can be calculated with equation (3.10). It is calculated from the parameters shown in figure 3.6, namely the chip height ($C_h$) that defines the minimum die to first row spacing, the bond pad length ($B_{PP, length}$), the bond pad to bond pad spacing of different chips ($B_{space}$), the substrate design rules such as via land ($D_{Via, land}$) and line space ($L_{space}$), and the factor $n$. This factor is 1 if the bondpad is escaped on one side by a via, 2 if the escaping vias are alternated or 0 when the bond pads are vias themselves.

Using several rows on the substrate needs a lot of rerouting space. The number of rows can be minimized by spreading the substrate bond pads. This is simple for ball-wedge bonding but has to be checked thoroughly for wedge bonding. The wedge tool has to be aligned for each pad to the bonding direction, which needs time. Additionally, the usual die bond pad for wedge bonding is long and small and thus limits non-orthogonal bonding. However, the die can be specially prepared for non-orthogonal bonding as described in [49]. The calculation of any angle bonding is described in subsection 4.1.5.2. Another limit is the bond wire length. An upper bound for this length ($L_{max}$) can be calculated with equation 3.11, assuming a wire-bond loop height of $H_{loop}$.

$$N_{Row} \leq \frac{B_{PP, length} \cdot C_{PP}}{H_{loop}}$$

(3.9)

$^2$Typical values are: $C_h = 450 \mu m$, $B_{PP, length} = 250 \mu m$, $B_{space} = 400 \mu m$.

$^3$Typical value for loop height is $200 \mu m$. 

---

**Figure 3.5:** Escaping area (gray) $A_{Esc}$ of a Component with its die size or pad area marked white.

**Figure 3.6:** Wire Bond Parameters
3.6. COMPONENT LIMITED SIZE

\[ O_{\text{edge}} = 2 \left( C_h + B_{\text{length}} + \frac{B_{\text{space}}}{2} \right) + (N_{\text{Row}} - 1) \left( B_{\text{length}} + n \cdot D_{\text{Via最主要的}} + L_{\text{size}} \right) \]

\[ L_{\text{size}} = \left( O_{\text{edge}} - B_{\text{space}} - B_{\text{length}} \right)/2 + C_h + 2 \cdot H_{\text{loop}} \]

3.5.2. Flip Chip

Flip Chip (FC) is the bonding technology that has the smallest overhead. Nevertheless, a spacing of 0.5 mm should be left between the dies for handling. Furthermore, one side needs larger space to allow underfilling the dies. For area array it is very important that the via pitch is smaller than the bump pitch to allow at least one line between two vias. Therefore, FC bonding can define the substrate design rules. Due to its minimal inductive and capacitive loads, FC is the bonding method with the best high speed performance.

3.5.3. Tape Automated Bonding (TAB)

TAB bonding adds the largest overhead to the die. It ranges from 2 to 20 mm. The larger sizes are I/O limited and can be calculated from the maximum number of chip pads \( C_N \) of one side, the wished OLB pitch \( P_{\text{OLB}} \) and the pad length \( P_{\text{Length}} \) using equation 3.12.

\[ O_{\text{TAB}} = 2 \left( C_N + P_{\text{OLB}} + P_{\text{Length}} + \frac{B_{\text{space}}}{2} \right) \]

3.5.4. Chip Size Packages

The footprint of a Chip Size Package (CSP) is mainly the package itself. Additionally, as they are placed with SMT placers, a spacing of about two mm for handling is needed around the component [50]. Beside having a rather large overhead, most CSPs are not well suited for highest speed as they often distribute peripheral ICs to area I/O adding distance and capacitive load.

In high volumes, CSPs are often less expensive than wire bonding due to the simpler assembly and the higher yield. But they suffer high initialisation cost and are more expensive than standard packages such as TSOP, QFP etc.

3.5.5. Bonding Options for Pentium MCM

The Pentium Processor reduced the bonding options drastically. With its minimum bond pad pitch of 75 \( \mu \text{m} \), flip chip was considered too challenging. TAB would ease manufacturing for the processor as it is already available in a TAB package. But this package is more than six times the die area as only one row for bonding on the substrate is possible. Furthermore, the MTSC with its two rows of pads on the die rend TAB difficult.

Thus, only wire bonding, which the dies were originally designed for, was left. Its footprint is shown in the next section.

3.6. Component Limited Size

First, the size of a module is estimated. The size is limited either by the space used by the components, the required routing area or the connection to the next level. We first estimate the
area needed by the components to derive the minimum size, the so called chip limited size. We will then examine in section 3.9 if this size is routable (i.e. we verify the assumption that the module is chip size limited).

If a module size is chip limited, its size depends on the selected chip packaging form (Single Chip Package, Chip Size Package or Bare Die) or the bonding method (TAB, Wire bonding, Flip Chip) as modeled in section 3.5. For analog circuits a fixed area is defined, because often a special layout and therefore area is required. Thus, the module size is estimated by adding up the component areas, their overhead, the analog areas and the required inter chip spacing. However, this does not take into account area lost between the ICs because the chip forms can not be changed to fit closely together. Some correct this lost area with a multiplier of 1.1 to 1.2, depending on the complexity. But a trial layout as shown in figure 3.7(a) is a much better estimation because it includes the lost area of the placement.

3.6.1. Pentium MCM

Table 3.3 shows a comparison of the estimation methods by assuming two rows of bonding, adding an overhead of 3 mm for the Pentium® processor, the MTSC and the two MTDPs. The others ICs require only one row of bonding (2 mm overhead). For the fanout on a microvia laminate additional 0.2 mm between the bond pad rows are necessary. It can be seen that the sum of areas is

<table>
<thead>
<tr>
<th>sum of areas silicon only</th>
<th>sum of areas with wire bonding</th>
<th>sum of areas incl. fan-out</th>
<th>trial layout wire bonding</th>
<th>trial layout + package comp</th>
</tr>
</thead>
<tbody>
<tr>
<td>420 mm²</td>
<td>735 mm²</td>
<td>800 mm²</td>
<td>31*31 mm²</td>
<td>32*32 mm²</td>
</tr>
</tbody>
</table>

24% smaller than the trial layout performed. Its estimated substrate size is 30.3*30.7 mm without connection to the next level, what corresponds to the final layout as shown in figure 3.7(b).

3.7. Power Distribution

The next step is the analysis of the power distribution. The goal is to estimate the required power, to find a strategy to distribute it within the module, to determine the required number of power pins and to decide how to decouple the power supply. Then the required routing resources for the power distribution can be calculated.

Power is either distributed with power planes or wired as single lines. Power planes have the advantage that signal lines have a controlled impedance and that they have a low power drop even for large currents. If the planes are very close to each other they provide a significant decoupling capacity.

Planes are easy to integrate in laminates as well as in cofired substrates. But in a thin film or photodefinition ceramic substrate, where every layer adds massively to the cost discrete, wiring might be preferred. Another alternative is mesh power routing[51] where on one layer alternatively a ground line, a signal line, a power line and again a signal line run parallel. The power is connected on the other layers by a 90 degree rotated mesh.

---

475 µm space/line, 150 µm via land
3.8. THERMAL ANALYSIS

The Module includes three different fixed power levels (2.9, 3.3, 5V). As only a few pins need 5V, it can be routed with lines instead of using a plane. The remaining two voltage levels can be supplied by a splitted plane because 2.9V is used by the Pentium only. Mesh routing was not considered because the power consumption with more than 2.5A at 2.9V and the interconnect requirements are so high that no layer could have been saved. Thus, the advantage of not having to structure the bottom plane would have been sacrificed.

Considering maximum power drop, bond wire size and length of the substrate to the package interconnect and maximum decoupling frequency of board level capacitors, the maximum power per bond wire was calculated as 100 mA. By using triplebonding, the number of power pins can be kept down to 56. In the implementation it was increased to 86.

The high speed decoupling is done on the module to avoid the inductivity of the substrate to package connection. To derive the type and number of capacitors for the Pentium, the Intel recommendations [52] were used. For the other dies, it was calculated from the maximum power drop allowed.

3.8. Thermal Analysis

As HDP solutions integrate the ICs onto a small area, the cooling has to be analyzed. Cooling requirements can reduce routing area when the path through the substrate has to be improved. After calculating the power dissipation of each component, the thermal path through the substrate and the path above the die should be analysed. One possibility to improve thermal conductivity is to include thermal vias into the substrate. However, this method reduces the routing resources. The path to the ambient depends on the cooling medium, its speed, the package and an optional heat spreader.

A simple analysis can be done using the thermal equivalent of ohmic resistors as described below.
3.8.1. Pentium MCM

The maximum power dissipation for the Pentium\textsuperscript{\textregistered} module is 16 W. This high power dissipation requires a detailed analysis of the thermal paths to avoid overheating of the ICs. Figure 3.8 shows the build-up and the thermal paths of a cavity down mounted package such as QFP or plastic stud grid array (PSGA). Due to this build-up, the main thermal path is from each IC through the adhesive, the dielectric, the substrate base and the package to the ambient. The other thermal paths are negligible.

The Pentium processor with its 7.5 W dissipates most of the power. Furthermore, the processor and the chip set have a rather low maximum junction temperature of 105 °C. By means of thermal vias through the substrate for these components, the junction to case temperature can be reduced by 7° to 5°. A worst case analysis of the main path using the thermal resistance model in figure 3.9 showed that a heat spreader and a fan connected to the backside of the package allows ambient temperatures of over 65° C.

![Image of Thermal Path of Pentium MCM](image)

**Figure 3.8: Thermal Path of Pentium MCM\textsuperscript{[44]} in a Ceramic QFP or Plastic Stud Grid Array**

3.8.2. Thermal Resistor Equivalent

The thermal path of the module in one dimension can often be easily modeled with the thermal equivalent of the ohmic resistors\textsuperscript{[53]} Here the equivalent to the electrical current is the heat to be dissipated. The voltage is then the temperature difference over the thermal resistor. These resistors depend on the transfer mode. In solid materials the heat is conducted (3.13) and it is convected into gaseous or liquid materials (3.14). These models are sufficient as long as the heat is conducted
along well defined paths that can be combined at metal layers i.e., the paths do not depend on spreading. The two modes can then be calculated as follows:

### 3.8.2.1. Thermal Conduction

Thermal conduction is the thermal mode within the package. It can be described as a thermal resistance depending on the conducting area \( A \), the thickness of the material \( d \) and its thermal conductivity \( k \) as shown in equation (3.13). Table 3.4 shows typical values for the thermal conductivity.

\[
R_{\text{Cond}} = \frac{A}{d \times k}
\]  

(3.13)

**Table 3.4: Thermal Conductivity of selected Materials**

<table>
<thead>
<tr>
<th>Material</th>
<th>( k ) ([W/m \times K])</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silicon</td>
<td>84</td>
</tr>
<tr>
<td>Silverfilled Glas</td>
<td>85</td>
</tr>
<tr>
<td>Polyimide-Quartz</td>
<td>0.35</td>
</tr>
<tr>
<td>Alumina</td>
<td>20</td>
</tr>
<tr>
<td>Aluminium</td>
<td>216</td>
</tr>
<tr>
<td>Copper</td>
<td>398</td>
</tr>
</tbody>
</table>

### 3.8.2.2. Thermal Convection

Thermal convection acts at the surface of a package. Its resistance depends on the surface \( A_{\text{Surface}} \) and the thermal conductivity \( h_c \) of the environment (3.14). The conductivity itself grows with the cooling medium’s speed. It can also be enhanced by a heat spreader. In table 3.5 typical values can be found.

\[
R_{\text{Conv}} = \frac{1}{h_c \times A_{\text{Surface}}}
\]  

(3.14)

**Table 3.5: Typical thermal Convection of selected Materials**

<table>
<thead>
<tr>
<th>Medium</th>
<th>( h_c ) ([W/(m^2 \times K)])</th>
</tr>
</thead>
<tbody>
<tr>
<td>still air</td>
<td>5</td>
</tr>
<tr>
<td>forced air</td>
<td>50</td>
</tr>
</tbody>
</table>

### 3.9. Routing Analysis

After the calculation of the routing requirements of the power distribution and thermal inserts, the substrate size defined by the interconnect requirements can be calculated. The sizes depend on the layer counts and the substrate design rules.
3.9.1. Required Interconnection Length

An approximation for the conductor length required to interconnect electronic components and devices, based on empirical analysis has been established by Seraphim [17]. This estimation is based on the following assumptions:

- the average interconnection length (Manhattan Distance) is 1.5\* die pitch ($P_D$) because an equal proportion of interconnections are made between nearest neighbours ($P_D$) and next to nearest neighbours ($2\*P_D$) and all other connections can be neglected.

- average fan-out of 1.5 per I/O

- Sum of all I/Os ($N_T$)

- wiring efficiency of 50% because the routing resources can usually not be used more efficiently.

- two pins are connected by one wire

The resulting average number of interconnections is multiplied by the average interconnection length and divided by the efficiency (3.15):

$$L = 1.5P_D \left( 1.5 \frac{N_T}{2} \right) / 50\%$$

(3.15)

The die pitch ($P_D$) can be estimated from a placement as shown in figure 3.7(a). Another possibility is to calculate $P_D$ as the average of all the component footprints lengths and widths plus the largest dimension and then divided by two. The calculation of $P_D$ can be difficult when interconnections to the outside are all around the substrate. In this case it can be neglected in the calculation if $N_T$ is reduced by the number of connections to the outside. For the analysis of double sided substrate, both sides have to be calculated independently. However, the interconnections that cross the substrate vertically need to be included in the pitch ($P_D$) and the number of pin ($N_T$) calculation.

3.9.2. Routing Density

The wiring pitch ($P_{w}$) obviously depends on the substrate technology. Often it is calculated as the sum of line width ($I_{width}$) and the minimum spacing between two lines ($L_{space}$). Figures for line pitches from literature [54] can be found in table 3.6. However, the effective wiring pitch not only depends on the minimum distance of two interconnection lines (line pitch) but is, to a large extent, dominated by the via pitch (which itself is a function of the via "land" area $D_{VIA LAND}$). Assuming that every second linespacing is governed by a via, the effective pitch can be calculated as the mean of the via pitch and the line pitch as shown in figure 3.10 and calculated in equation (3.16).

$$P_{wire} = \frac{I_{width} + L_{space}}{2}$$

(3.16)

\[^5\text{calculated using 3.16 with } I_{width} = 20\mu m, D_{VIA LAND} = 50\mu m \text{ and } L_{space} = 30\mu m\]
3.9. Routing Analysis

Table 3.6: Wiring Pitch \( (P_w) \) for Different Substrate Technologies

<table>
<thead>
<tr>
<th>( P_w )</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.150mm</td>
<td>SBU Laminate</td>
</tr>
<tr>
<td>0.250mm</td>
<td>Cofired Ceramic</td>
</tr>
<tr>
<td>0.050mm</td>
<td>Thin Film</td>
</tr>
<tr>
<td>0.065mm</td>
<td>Thin Film IMC(^3) (calculated)</td>
</tr>
</tbody>
</table>

![Figure 3.10: Design rules and proposed Wiring Pitch \( (P_w) \)](image)

3.9.3. Layer vs. Area

The necessary overall wiring area \( (A_W) \) can be estimated from the total interconnection length \( (L) \), the wiring pitch \( (P_w) \) and the number of interconnection layers \( (N_{Layer}) \) using equation (3.17):

\[
A_W = \frac{L \times P_w}{N_{Layer}} \quad (3.17)
\]

By means of this equation a trade-off between the number of layers and the substrate technology i.e. the wiring pitch can be done. This trade-off can be done with figure 3.11. It shows the required number of signal layers as a function of the desired substrate size for all substrate technologies in table 3.6.

3.9.4. Pentium MCM

The graph in figure 3.11 is drawn for the Pentium example with a mean chip pitch \( P_D = 12 \text{ mm} \) from the trial placement in figure 3.7(a) and the number of pins \( N_T = 1015 \). In this example the chip limited size is routable on a thin film substrate on two signal layers. Using the laminate SBU technology, four build up layers are required. For a ceramic implementation a theoretical seven layer substrate would be required. However, this has been considered impossible due to the exponential increase of vias and the escaping overhead from the bond pads to inner layers which would lead to wider spreading of the bond pads. By extending the area for the laminate to 38 \( \times \) 38 mm, one layer is saved and the substrate becomes less expensive. Therefore, there remain two implementation options, namely a 32\( \times \)32 mm thin film and a 38\( \times \)38 mm laminate SBU solution.
3.10. Signal Distribution

Signal integrity on the interconnections as well as switching noise on the power nets have to be investigated to ensure proper functionality of the whole system.

3.10.1. Pentium MCM

Figure 3.12 shows a SPICE simulation of the address bus. This analysis of the impedance controlled 20μm lines showed that, due to the high resistivity (500 Ω/m) of the thin film lines, the signals have a good quality. The signal distortion from the capacitive loads (chip I/Os) as well as the transmission lines themselves is minimal. This good signal quality is even valid for the rather long (about 10 cm), 66 MHz clocked address lines of the SRAM ICs without any termination. In the graph in figure 3.12 the pentium drives these address lines. The risetime degrades from internal 2 ns to external 3 ns starting at t=2 ns. Furthermore, the skew between all the ICs on the address bus is smaller than 1 ns. This signal quality is much better than on a PCB featuring packaged components.

3.11. Design for Test

The aim of the partitioning was to keep internally as many nets as possible. Therefore, they can not be contacted for test purposes. Furthermore, a conventional in-circuit test that contacts test points by means of nails fails because of the fine structures and the tight pitches. After all, a MCM combines the complexity of a fully assembled system with the access limitations presented by integrated circuits. To overcome these limitations Design for Test (DFT) is crucial for the design process. To select a suitable strategy, table 3.7 compares possible test approaches.
3.11. Design for Test

![Graph showing SPICE Simulation of SRAM Address Line]

*Figure 3.12: SPICE Simulation of SRAM Address Line showing the good signal quality.*

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional Testing</td>
<td>Complete test</td>
<td>Writing test vectors is time consuming</td>
</tr>
<tr>
<td></td>
<td>System test</td>
<td>and complex</td>
</tr>
<tr>
<td></td>
<td>At-speed test</td>
<td>Often no simulation models and test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vectors of ICs available</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Difficult to localize assembly level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>faults</td>
</tr>
<tr>
<td></td>
<td>No extra hardware in the module required</td>
<td>Expensive test infrastructure (VLSI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>test system)</td>
</tr>
<tr>
<td>Assembly Testing</td>
<td>Simple test vectors</td>
<td>No functional test</td>
</tr>
<tr>
<td>1. provided test</td>
<td>Simple tester sufficient</td>
<td>Not provided by all chips</td>
</tr>
<tr>
<td>structures</td>
<td>No extra cost</td>
<td></td>
</tr>
<tr>
<td>2. additional test</td>
<td>Good fault coverage</td>
<td>Cost</td>
</tr>
<tr>
<td>structures</td>
<td>At-speed test</td>
<td>Complexity of the added circuitry/test</td>
</tr>
<tr>
<td></td>
<td></td>
<td>structure</td>
</tr>
<tr>
<td>In-Circuit-Test</td>
<td>At-speed test</td>
<td>No system test</td>
</tr>
<tr>
<td></td>
<td>Same vectors as for single parts</td>
<td>100% accessibility to all chips necessary</td>
</tr>
<tr>
<td></td>
<td>Extended test of each die</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>All busses must tri-state outputs</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Difficult for HDP</td>
</tr>
</tbody>
</table>

### 3.11.1. Pentium MCM

Due to the high complexity of the Pentium MCM, which includes more than hundred internal nets and about 1/00 wire bonds, a powerful test strategy is mandatory. Mainly a manufacturing test is important to locate faults early in the manufacturing flow. It can be combined with an at speed test
and a final functional test to guarantee the same test level as for standard components [55].
To overcome the access limitations, JTAG, Nand-Tree mode as well as about 60 designated test pins are used to test the connection between the ICs. Furthermore, these features allow functional and speed testing of the individual chips if the JTAG of the Pentium works. The tests and characterisation of this module were done on a HP-83000 ASIC tester.

3.12. Package Selection

The package protects a module from mechanical, thermal and electrical influences and light, and connects the module to the outside. Depending on the requirements, the following technologies can be used:

- Standard package
- Connector
- Ball on substrate bottom or BGA package
- Cable

An earlier selection was not possible because the number of test pins is an important parameter. The number of pins can now be calculated as the sum of the number of signal, power and test pins. The substrate size defines the cavity size of the package. These two parameters together with further constraints such as thermal conductivity etc. now allow to choose or design a suitable interface to the motherboard.

3.12.1. Pentium MCM

For the Pentium MCM more than 300 pins were required and the cavity size with its 33*33 mm was much larger than for a single chip. Therefore, we could not use an available standard package. We did not want to use connectors, as the expensive substrate would have become larger. A cable interface was considered impossible due to the high I/O count and their speed of 33 MHz.

BGAs have the advantage of providing a lot of pins on a small area and a good thermal path. However, for thin film substrate it is difficult to mount balls on the backside. This configuration would conduct all the heat into the motherboard from where it could not completely be dissipated. Thus, a complicated thermal connection to the active IC side would be required.

To fulfill the thermal requirements, a package where the substrate was mounted cavity down was chosen. Thus, a heat spreader can directly be connected to the package's backside with a good thermal path to the substrate's backside and the ICs as shown in figure 3.8.

For the prototypes a ceramic QFP featuring 308 pins and a very large cavity was used. This too large cavity was bridged by means of an additional substrate. For the pre-series a plastic stud grid array (PSGA) [56] with integrated metal back was developed as shown in figure 3.13. The PSGA is an injection moulding package. Its pins are small metallized plastic studs. It can be mounted to a motherboard like a BGA.

3.13. Cost and Yield

When all the technical aspects are considered and several alternatives have been generated, yield and cost estimates can be made based on the parameters calculated in the feasibility analysis.

To estimate substrate and assembly cost, it is best to get offers from manufacturers because there are hardly any standard values and they are only very rough estimations as each HDP solution requires other features. For a quotation a manufacturers requires.
Design rules, substrate layer count, size and surface finish
- Number of bare dies and its types, number of wire bonds and other components
- Packaging such as glob top, cap and BGA balls

It is very difficult to get yield data. As a rule of thumb the following values can be used:
- Substrate (interconnects only and electrically tested) 99%
- Packaged components 99.9%
- Bare dies 90 - 99% (depending on their test level)
- Wire bond 99.99 - 99.999% per bond

The cost for a working module can now be calculated with equation 3.18:

$$\text{cost} = \sum_{k} \frac{\text{direct costs}}{\Pi \text{yields}}$$  \hspace{1cm} (3.18)

This rather simple approach is sufficient if an existing infrastructure is used. If the manufacturing process can be optimized, a more detailed analysis as described in [57] is required.

**3.13.1. Pentium MCM**

For the processor module, offers from manufacturers all over Europe showed that a laminate with SBU can be more expensive than the thin film solution provided by IMC. Table 3.8 shows that a technology is not competitive any more if used at its limits. This happened here with the laminate technology.

The preceding sections described the required steps to generate alternatives and to extract their features. Alternatives not fulfilling the specification are already sorted out. If the cost is not the only decision criteria it can be weighted with other parameters such as size, speed etc. Often customers are ready to pay slightly more for a smaller module, especially when it saves them a daughterboard or reduces the size of an external housing. For the decision a figure of merit can be calculated. In this example in table 3.8 it is 1 divided by size (in m²)*cost, but it depends on individual weights.

3.14.1. Pentium MCM

<table>
<thead>
<tr>
<th>Technology</th>
<th>size</th>
<th>thermal conductivity</th>
<th>cost normalized</th>
<th>figure of merit [1/size*cost]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laminate with SBU</td>
<td>1444</td>
<td>bad</td>
<td>1.5</td>
<td>460</td>
</tr>
<tr>
<td>MCM-D IMC</td>
<td>1024</td>
<td>excellent</td>
<td>1</td>
<td>980</td>
</tr>
</tbody>
</table>

The remaining options for the Pentium MCM laminate or thin film substrate with wire bonding mounted into a package are equivalent in all features but size, thermal conductivity and cost. The thermal conductivity of laminates can be improved for example by means of thermal vias. Therefore, only size and cost where considered for this example. As shown in table 3.8 the thin film solution is smaller and has a lower price. Therefore, considering the figure of merit, this solution was chosen.

3.15. Summary

As illustrated in figure 3.3, first all technical aspects where considered. Finally, a cost-performance evaluation chose one option for implementation. Therefore, only one solution has to be layouted and its technology and its placement are already fixed.

The modelling in this chapter was simple and often based on rules of thumb. They are good for a first order estimation and to understand the methodology. To enable a precise evaluation, parametric models for the substrate size are described in the next two chapters 4 and 5 followed by the description of their implementation in chapter 6.
4

Component Footprint and Chip Limited Substrate Size

The relative simple models used in chapter 3 are good for a first order estimation. In a second pass the models or tools presented in this chapter can be used to generate viable build-up alternatives. They mainly cover bonding, size and escape routing more accurately but require more input, too. These models are also the basis to automate the feasibility analysis as well as the layout.

The footprint of a component using any first level interconnect or package and mounted on whatever substrate technology is larger than the pad area as described in section 4.1. Some examples are then presented in section 4.2. After the calculation of the component's footprints, the chip limited substrate size can be estimated as described in section 4.3.4.

The models presented in this chapter allow an early estimation of the footprint with minimal input data. To estimate the impact of the package type on the system size, our models calculate the footprint from given package descriptions (size, pitch and number of I/Os). They use a library where the available substrate technologies are specified. Furthermore, the number of signal layers needed to escape a component is calculated and escape layout can be done automatically.

These models are based on a lot of equations and parameters. A description of the parameters can be found in appendix A. With these parameters the footprint can be calculated as described in the section below. In some equations a value is estimated to reduce the number of input parameters. In these cases the estimated values might be replaced by the effective one as marked in the equations with "real number".

4.1. Footprint

As defined below the footprint is more than the component size itself, because the minimum distance between adjacent components has to be added. This distance depends on the needed spacing for pick and place tools and/or the chosen interconnect. For lower density substrate the footprint is defined by the needed escape routing.

4.1.1. Definition

I define the footprint of a package

1. as the pure component area, i.e. its size:
Chapter 4: Component Footprint and Chip Limited Substrate Size

Figure 4.1: Escaping from pads (black) on a PCB with 125 \( \mu m \) line/space and 650 \( \mu m \) via land diameter (unfilled circles) for QFP 100, BGA 100.

2. plus the maximum of
   - either assembly keepout overhead, e.g. for underfiller, the SMT placer etc;
   - or I/O escape overhead, i.e. additional space for escaping all chip I/Os to another layer (see Fig.4.2).

Whereas the component size and the assembly keepout overhead are easy to determine, the I/O fanout overhead is not simple to calculate. Escaping from the pads to the vias can consume significant area. Thus, it can dominate the spacing needed between two adjacent components. Moreover,

Figure 4.2: Overhead for single 100 \( \mu m \) wide line escaping (black) from area I/O with 0.65 \( mm \) pad pitch to vias (unfilled circles) with 350 \( \mu m \) diameter and 100 \( \mu m \) spacing (gray)
this escape area is highly dependent on substrate technology and design rules as shown in figure 4.1. A 100-pin BGA with 1 mm/IO pitch is much smaller than a QFP, but it consumes large escaping area, so the package footprint is 60% larger than the package itself. This is due to the fact that each trace has to leave the top layer with a via to an inner layer in order to place components as dense as possible. The escape routing not only needs area but can also block a number of layers under and next to the component for escaping from inner pads. It can define the number of layers needed even if the global routing could be done on less layers as modelled in the remainder of this section.

4.1.2. Modelling

As defined above, the footprint is the maximum of the assembly keepout area or the area including the escaping ($A_{Eca}$). Whereas the assembly keepout area is easily calculated as the component size (A) plus the minimum assembly spacing needed between adjacent components, the escaping area is the area needed for the components pads ($A_{Pad}$) plus the area needed by the escaping overhead for the escaping vias on ($O_{Esc}$) as shown in equation (4.1) and figure 3.5.

$$A_{Eso} = \left(\sqrt{A_{Pad}} + O_{Esc}\right)^2$$

The escaping overhead depends on the pad pitch, the design rules and the ball locations as detailed and modelled in the subsections. The models for the effective footprint are based on the following principles:

4.1.2.1. Number of Layers

To avoid local congestions, it is very important to know the minimum number of layers required for escaping. It defines the number of layers needed if global routing could be done on less layers. In addition to the inner layers, the top layer can also be used for escaping even when in reality the top layer is often left unrouted to allow for dense component placement.

4.1.2.2. Escaping

For the area usage each trace is modeled to leave the top layer with a via to an inner layer. For components or interconnects that need several pad rows, such as wire bond, area flip chip and BGA, on each layer the fanout of as many rows as possible is done in order to minimize the number of layers needed. Best case is when the routing density is so high that all rows can escape between the pads of the outermost row. Otherwise inner rows have to stagger down to additional layers.

The escaping of the outermost row is always routed to the outside of a component. Inner wire bond rows or single pad rows can be escaped to both sides. Area pads are routed to the outside on a minimum number of layers. If the array center is depopulated, the innermost rows can be escaped to this free space. This is important when the pad pitch is too small to place vias between the pads.

When the via lands are larger than the pads and escaping alternating to both sides is not possible or not sufficient, vias can be arranged in rows as illustrated in figure 4.3. The effectively necessary area depends on the via type (through hole or staggered) and if vias are allowed in pads. For the calculation of the escape area the vias are modelled to escape on one side of a pad row with $k$ being the number of vias in one row. Between two vias of adjacent via rows the following number of lines has to pass.
Figure 4.3: Escaping from wire bond pads (black) to staggered vias (infilled circles) on the top and to through hole vias on the bottom on the top layer (black lines) and to the outside on an inner layer (dark gray). The minimum spacing is marked grey. Wire bond pad pitch is 225 and 300 µm, line/space is 75/125 µm, the small vias are staggered microvias with a land diameter of 500 µm and the larger through hole vias have a land diameter of 650 µm.

- Through hole vias: \( k \) lines

- Staggered vias: \( k - 1 \) lines for the last via and \( k - 2 \) lines between diagonal vias.

The distance between two vias in one row (\( d_{Via} \)) depends on the number of lines between the vias. For staggered vias or microvias there is no need for lines between. Thus, the minimum distance should be used (4.2):

\[
d_{Via-stagger} \geq D_{ViaLand} + L_{Space}
\]  

(4.2)

Through hole vias on the other side would block the whole area underneath the escaping. So, \( L_{EscVia} > 0 \) lines should pass between the vias as illustrated in figure 4.3 and as calculated in formula 4.3:

\[
d_{Via-through} = D_{ViaLand} + L_{Space} \]

\[+ L_{EscVia} \times (L_{Space} + L_{Width})\]

(4.3)

For both via types the number of vias in one row (\( k \)) has to be calculated first. \( k \) is also the number of pads that are connected to one via row and depends on the via land diameter (\( D_{ViaLand} \)), the wiring pitch and the Pitch. The Pitch is the bondpad pitch (\( C_{PP} \)) or the ball/bump pitch (\( P_{Boll} \)) divided by the number of lines passing between the ball/bumps (\( l_{top-eff} \)). For through hole vias \( k \) is calculated so that \( k \) lines can pass between the vias (4.4). Staggered vias can be put closer together as only \( k - 1 \) lines have to pass between as calculated in equation (4.5):

\[
k_{through}(Pitch) \geq \frac{D_{ViaLand} + L_{Space}}{Pitch - L_{Space} - L_{Width}}
\]  

(4.4)
4.1. Footprint

\[ k_{\text{stagger}}(\text{Pitch}) \geq \frac{D_{\text{ViaLand}} - L_{\text{Width}}}{\text{Pitch} - L_{\text{Space}} - L_{\text{Width}}} \]  \hspace{1cm} (4.5)

This escaping strategy is only possible if the escape pitches are much larger than the line pitch. Otherwise, \( k \) would become very large.

Before we can calculate the escaping overhead, the escaping distance per via except the first one (\( d_{E_{\text{sc}}} \)) is then calculated with (4.6) for through hole and (4.7) for staggered vias. These formulas are geometrically calculated from the number of lines that have to pass between two rows as illustrated in figure 4.3.

\[
\begin{align*}
    a_{\text{through}} &= \left[ D_{\text{ViaLand}} + k \cdot L_{\text{Width}} + (k + 1) \cdot L_{\text{Space}} \right]^2 \\
    a_{\text{staggered}} &= \left[ D_{\text{ViaLand}} + (k - 2) \cdot L_{\text{Width}} + (k - 1) \cdot L_{\text{Space}} \right]^2 \\
    d_{E_{\text{sc-through}}} &= \sqrt{d_{V_{\text{via}}}^2 - \left(\frac{a_{\text{through}} - k \cdot \text{Pitch}^2}{2 \cdot k \cdot \text{Pitch}}\right)^2} \\
    d_{E_{\text{sc-staggered}}} &= \sqrt{d_{V_{\text{via}}}^2 - \left(\frac{a_{\text{staggered}} - k \cdot \text{Pitch}^2}{2 \cdot k \cdot \text{Pitch}}\right)^2} 
\end{align*}
\]  \hspace{1cm} (4.6) \hspace{1cm} (4.7)

When the parameters \( k \) and \( d_{E_{\text{sc}}} \) are calculated, the total escaping overhead is estimated using (4.8). Or if vias are allowed in the substrate pads, the overhead is smaller, as shown in formula 4.9:

\[
\begin{align*}
    O_{E_{\text{sc}}-\text{NoVias}} &= 2 \cdot \left[ D_{\text{ViaLand}} + 2 \cdot L_{\text{Space}} \right] + (k - 1) \cdot d_{E_{\text{sc}}} \\
    O_{E_{\text{sc}}-\text{ViasInPads}} &= 2 \cdot \left[ (k - 1) \cdot d_{E_{\text{sc}}} + L_{\text{Space}} \right] 
\end{align*}
\]  \hspace{1cm} (4.8) \hspace{1cm} (4.9)

![Figure 4.4: Escaping from wire bond pads (black) to vias (unfilled circles) on both sides of a row. In the upper part staggered vias are shown and in the lower part the escaping is to through hole vias.](image)

For peripheral connections such as QFPs, wire bond and TAB, the vias can be escaped to both sides of a row as shown in figure 4.4. In this case \( k1 \) pads are escaped to one via row to the inside.
The equations for \( k \) (4.5, 4.4) remain the same but \( k \) is divided into \( k_1 + k_2 \). The escaping per via \( (d_{Esc}) \) remains the same for through hole vias. But it is reduced to \( d_{Esc2} \) for staggered vias because less lines have to pass between the vias. Its calculation depends on the shortest distance to the adjacent via row which is parameterized with \( n \). \( n + 1 \) is the minimum number of vias in a row needed to span a triangle with the first via of the adjacent row as shown in figure 4.4 and calculated in (4.10). When \( n \) is not smaller than \( k_1 \), then \( d_{Esc2} \) becomes zero because all vias on the inner side fit next to each other. Otherwise it is calculated in equation (4.11).

\[
\begin{align*}
    n &> \frac{k \cdot pitch - (k_1 - 1) \cdot L_{\text{Width}} - k_1 \cdot L_{\text{Space}} - D_{\text{ViaLand}}}{d_{\text{Via}} - L_{\text{Width}} - L_{\text{Space}}} \quad (4.10) \\
    h &= \frac{(n^2 d_{\text{Via}}^2 + k^2 \cdot Pitch^2 - [D_{\text{ViaLand}} + (k_1 - n - 1) \cdot L_{\text{Width}} + (k_1 - n) \cdot L_{\text{Space}}]})}{(2 \cdot k \cdot Pitch)} \\
    d_{Esc2} &= \frac{\sqrt{\frac{n^2 d_{\text{Via}}^2 - h^2}{n}}}{\mu} \text{ if } n < k_1 \text{ else } d_{Esc2}(Pitch) = 0 \quad (4.11)
\end{align*}
\]

### 4.1.2.3. Escaping from Area Pads

For area I/O the number of rows that could be routed on the top layer \( (l_{top}) \) is the same as the number of lines between bumps/balls \( (l_{Pads}) \) plus one. \( l_{Pads} \) depends on the pad diameter of the bumps/balls on the substrate \( (P_{Width}) \) (4.12). The analog parameter for the inner layers \( (l_{in}) \) is calculated as the number of lines between two vias \( (l_{Vias}) \) placed in the same pitch as the bump/ball pitch \( (P_{Pitch}) \) plus one if inner rows can be escaped to another layer, otherwise it is 0 (4.13). Inner rows can be escaped if vias can be placed either into the pads or between them.

\[
\begin{align*}
    l_{Pads} &\leq \frac{P_{Pitch} - P_{Width} - L_{Space}}{L_{Width} + L_{Space}} \\
    l_{top} &= l_{Pads} + 1 \quad \text{(4.12)} \\
    l_{Vias} &\leq \frac{P_{Pitch} \cdot D_{\text{ViaLand}}}{L_{Width} + L_{Space}} \text{ or } 0 \\
    l_{in} &= l_{Vias} + 1 \text{ or } 0 \quad \text{(4.13)}
\end{align*}
\]

### 4.1.2.4. Escaping for high values of \( k \)

For high values of \( k \) the presented models for the escaping vias are over pessimistic because the space in the corner of a component is not considered for via placement. The calculation of escaping over head \( (O_{Esc}) \) neglects that the corners of the escaping area can also be used to place vias as shown in figure 4.5. To avoid this effect Bruhin [58] developed a model based on my work.

For \( k \) larger than 4 and pads on all four sides th the vias can be placed in a trapezoid as shown in figure 4.6. The vias are placed in rows parallel to the chip edge. Their pitch \( d \) allows a line to pass between the vias (4.14). The same applies for the first row. With these constraints the available row width \( (W_n) \) is calculated in equation (4.15). Here a spacing of \( D_{\text{ViaLand}} / 4 \) on both sides is left empty to avoid design rule violations from the neighbouring trapezoid.

\[
d = D_{\text{ViaLand}} + 2 \cdot L_{\text{Space}} + L_{\text{Width}} \quad \text{(4.14)}
\]
4.1. Footprint

\[
W_n = \frac{\text{Package}_{\text{length}}}{\text{Package}_{\text{width}}} + 2 \cdot \frac{\text{Package}_{\text{length}}}{\text{Package}_{\text{width}}} + (d + n + \frac{D_{\text{Vialand}}}{4} + 2 \cdot L_{\text{Space}} + L_{\text{Width}})
\]

(4.15)

In this width \(W_n\), \(E_{\text{in}a}\) pads are connected from the inside on the upper layer (black in figure 4.6) and \(E_{\text{out}a}\) are escaped on the layer below (dark gray). The number of rows \(n\) is increased until \(E_{\text{in}a}\) becomes zero. For the first row all the pads are escaped and no line has to pass between the vias on the second layer as shown in equations (4.16ff). \(E_n\) is then reduced respectively enlarged by the number of vias in a row \(V_n\) in equations (4.19ff).

\[
E_{\text{in}-\text{Wirebond}} = C P S / N_{\text{row}} - V_0 \quad (4.16)
\]

\[
E_{\text{in}-\text{Other}} = C P S - V_0 \quad (4.17)
\]

\[
E_{\text{out}_0} = 0 \quad (4.18)
\]

\[
E_{\text{in}_a} = E_{\text{in}_a-1} - V_0 \quad (4.19)
\]

\[
E_{\text{out}_a} = E_{\text{out}_a-1} + V_0 \quad (4.20)
\]

For the first rows \(E_{\text{in}}\) is dominant whereas for the outer rows \(E_{\text{out}}\) defines the number of lines that have to pass between the vias as shown in figure 4.6 and in equation (4.21).

\[
E_n = \max(E_{\text{in}_a}, E_{\text{out}_a}) \quad (4.21)
\]

\[
V_n - \text{staggered} \leq \frac{W_n - E_n \cdot L_{\text{Pitch}} - L_{\text{Space}}}{D_{\text{Vialand}} + L_{\text{Space}}} \quad (4.22)
\]

\[
a = \max(E_n - V_n - \text{through} \cdot L_{\text{Vialand}}; 0) \quad (4.23)
\]

\[
V_n - \text{through} \leq \frac{W_n - a \cdot L_{\text{Pitch}} - L_{\text{Space}}}{D_{\text{Vialand}} + L_{\text{Space}} + L_{\text{Vialand}} \times L_{\text{Pitch}}} \quad (4.23)
\]

To calculate the overhead with the standard equations (4.8f) the parameters \(k\) and \(d_{\text{Esc}}\) are derived from the number of escaping rows \(n_{\max} + 1\) and their distance \(d_{\text{Esc}}\) as calculated in equations (4.24f).

\[
k_{\text{eff}} = n_{\max} + 1 \quad (4.24)
\]

\[
d_{\text{Esc}} = d + L_{\text{Pitch}} / n_{\max} \quad (4.25)
\]
4.1.3. TAB

TAB mounting has its pads arranged in one row. This can cause a large footprint for high pin count ICs. In this case the area for the pads \( A_{Pads} \) depends on the outer lead bonding pitch \( P_{OLB} \) and the pad length \( P_{Length} \) and can be calculated using (4.26). If only a few pads are connected with TAB as first level interconnect, the dominating factor is the minimum distance from chip to pad \( C2P_{TAB} \) as shown in equation (4.27). Thus, the maximum of both has to be used. On the other hand the single pad row allows to fanout the row to both the in- and the outside of the component. Thus, \( k \) is reduced by \( k_1 \) via rows that fanout to the inside to reduce the escaping area. This changes (4.8) to (4.28).

\[
A_{Pads} = \left( \frac{C_N}{A \text{ or real number} \times P_{OLB} + 2P_{Length}} \right)^2 \quad (4.26)
\]
\[
A_{Pads} = \left( \sqrt{A} + 2 \times (C2P_{TAB} + P_{Length}) \right)^2 \quad (4.27)
\]
\[
O_{Esc} = 2 \times (D_{ViaLand} + 2 \times L_{Space} + (k - 1 - k_1) \times d_{Esc}) \quad (4.28)
\]

The missing parameter \( k_1 \) can either be chosen or calculated iteratively according to the model shown in figure 4.7. Distance \( l \) is the space available for placing the vias. It depends on the difference between the pad area \( A_{Pads} \) and the component size \( A \), the pad length \( P_{Length} \) and the minimum distance from the die required for die bonding \( C2P_{min} \). In this space a maximum of \( n_{max} \) vias can be placed at the pitch \( d \) is greater or equal than a via land plus one line passing between two rows. From these parameters equation (4.29) calculates the available row width \( W_n \) which is used in (4.30) to calculate the number of vias per row \( (V_n) \) for staggered vias. Finally \( k_1 \) is calculated in 4.31. Here, the number of vias \( (E_{n_{max} - 1}) \) is divided by how many times \( k \) pads are on one side of an IC \( (= C_N / A \text{ pads}) \).

\[
l = \frac{\sqrt{A_{Pads}} - \sqrt{A}}{2} = P_{Length} - C2P_{min}
\]
\[
n_{max} \leq l/(D_{ViaLand} + 2L_{Space} + L_{Width})
\]
\[
d = l/n_{max}
\]
\[
W_n = 2 \times \frac{l_2}{l_1} \times (d \times n + \frac{D_{ViaLand}}{2}) + \sqrt{A} + 2 \times C2P_{min} \quad (4.29)
\]
\[
V_n - stagger \leq \frac{W_n - 2 \times L_{Pitch}}{D_{ViaLand} + L_{Space}}
\]
\[
E_{n_{max} - 1} = 0
\]
For through hole vias $E_{\text{EscVias}}$ lines have to pass between the vias to avoid the blocking of all layers. Thus (4.30) becomes (4.32). Unfortunately, (4.32) has no closed form solution but has to be calculated iteratively.

\[
\begin{align*}
E_n & = E_{n-1} + V_n - 1 \\
V_n - \text{stagger} & \leq \frac{W_n - 2 \times L_{\text{Pitch}} - E_{n-1} \times L_{\text{Pitch}}}{D_{\text{ViaLand}} + L_{\text{Space}}} \\
kl & \leq \frac{E_{n-1} - a}{4k}
\end{align*}
\]

\[(4.30)\]

\[
\begin{align*}
W_0 - \text{through} & \leq \frac{W_0 - 2 \times L_{\text{Pitch}}}{D_{\text{ViaLand}} + (i + 1) \times L_{\text{Space}} + i \times L_{\text{Width}}} \\
\alpha & = \max(E_{n-1} - V_n; 0) \\
V_n - \text{through} & \leq \frac{W_n - 2 \times L_{\text{Pitch}} - \alpha \times L_{\text{Pitch}}}{D_{\text{ViaLand}} + (i + 1) \times L_{\text{Space}} + i \times L_{\text{Width}}}
\end{align*}
\]

\[(4.32)\]

Whereas TAB and wirebond are well suited for peripheral IC pad configurations, Flip Chip allows to pads over the whole area of an IC as described in the subsection below. Furthermore, TAB can also be used to reroute the peripheral connections to an area configuration as described in subsection 4.1.6. An IC like that is called chip size package.

### 4.1.4. Flip Chip

Flip Chip has the smallest footprint if only the first level interconnect is considered. But it concentrates the I/Os on a small area. These many I/Os need an area for escaping. Additionally, the flip chip bump pitch ($P_{\text{Pitch}}$) has to be larger than the routing pitch to allow escaping from inner rows for area I/O.
4.1.4.1. One Row of Bumps

For one on-chip row of bumps the escape vias can be added on one ($k_1 = 0$) or both sides of the row. For large vias the escape area can be large as illustrated in figure 4.3 for wire bond pads. Escaping only to the outside is preferred because vias under the IC disturb the flow of the underfill. To avoid the disturbance the vias also could be filled before FC mounting. The minimum bump pitch is therefore equivalent with the line pitch or somewhat larger for $k_1 = 0$. $k$ is calculated with formula (4.5) or (4.4) replacing $Pitch$ with the bump pitch ($P_{\text{Pitch}}$). The overhead is the escaping of the bumps (4.8f) or the spacing from die to die ($FC_{\text{Spacing}}$) (4.33). If the escaping is to one side ($k_1 = 0$) only the top layer is used. Otherwise two layers are required (4.34).

$$\begin{align*}
V_{\text{inPads}} & = \text{Allowed and } P_{\text{Width}} \geq D_{\text{ViaLand}} \\
O_{\text{Esc-FC1Row}} & = O_{\text{Esc}}(Pitch = P_{\text{Pitch}}) \\
N_{\text{Layer}} & = 1 \text{ for } k_1 = 0 \text{ else } 2
\end{align*} \ (4.33)$$

$$N_{\text{Layer}} = 1 \text{ for } k_1 = 0 \text{ else } 2 \ (4.34)$$

4.1.4.2. Full Area Array

To cover not only quadratic arrays, the long and the small sides are parameterized as $N_{\text{Layer}}$ for the number of rows on the larger side, and $N_l$ for the number of rows on the smaller side. On the top layer ($2\times l_{\text{Top}}$) rows and $4\times (l_{\text{Top}} - 1)^2$ pads can be escaped on the top layer. Additional four bumps are escaped if lines can pass between the pads ($l_{\text{Top}} > 1$) as expressed by the parameter $e_{\text{Top}}$. If inner bumps can be escaped, the same applies for the inner layers with the parameters $l_{\text{in}}$ and $e_{\text{in}}$. The power bumps ($P_{\text{P}}$) can be subtracted from the number of bumps to escape. But $P_{\text{P}}$ does not include the power bumps in the outermost row because they are routed on the top layer anyhow. Thus, the number of remaining bumps is given by:

$$N_{\text{RemTop}} = (N_l - 2l_{\text{Top}})(N_s - 2l_{\text{Top}}) - 4\times (l_{\text{Top}} - 1)^2 - if(l_{\text{Top}} > 1; 4; 0) - P_{\text{P}} \ (4.35)$$

A component can be escaped if all its bumps are escaped on the top layer or if inner rows can be escaped by vias. Based on equation (4.36) the number of layers is calculated with (4.37).

$$e_{\text{Top}} = if(l_{\text{Top}} > 1; 4; 0), \ e_{\text{in}} = if(l_{\text{in}} > 1; 4; 0)$$

$$N_{\text{Rem}} = (N_l - 2l_{\text{Top}} - 2l_{\text{In}}) \times (N_s - 2l_{\text{Top}} - 2l_{\text{In}})$$

$$-4\times (l_{\text{Top}} - 1)^2 - 4n(l_{\text{in}} - 1)^2 - e_{\text{Top}} - n\times e_{\text{in}} - P_{\text{P}} \ (4.36)$$

$$a = 4\times l_{\text{in}}^2$$

$$b = -2\times l_{\text{in}} \times (N_l + N_s - 4l_{\text{Top}} - 4 + 2l_{\text{in}})$$

$$c = N_l \times N_s - 2\times l_{\text{Top}} \times (N_l + N_s - 4) - 4 - e_{\text{Top}} - P_{\text{P}}$$

$$n \geq \frac{-b - \sqrt{b^2 - 4\times a \times c}}{2 \times a} \ (4.37)$$

$$N_{\text{Layer}} = n + 1$$

When inner layers are needed for escaping, less than the maximum possible number of rows ($l_{\text{Top}}$) may be escaped on the top layer (4.38). Finally, the escaping size is given by (4.39).

$$l_{\text{Top-eff}} \geq (4n^2(l_{\text{in}})^2 - 2nl_{\text{in}} \times (N_l + N_s - 4 + 2l_{\text{in}})$$

$$-4n - 4 + N_l \times N_s - N_{\text{P}} - n\times e_{\text{in}} - e_{\text{Top}})$$

$$f(2\times (N_l + N_s - 4 - 4nl_{\text{in}}))$$

$$A_{\text{Esc}} = ((N_l - 1) \times P_{\text{Pitch}} + P_{\text{Width}} + O_{\text{Esc}})$$

$$\times ((N_s - 1) \times P_{\text{Pitch}} + P_{\text{Width}} + O_{\text{Esc}}) \ (4.38)$$

$$A_{\text{Esc}} = \frac{((N_l - 1) \times P_{\text{Pitch}} + P_{\text{Width}} + O_{\text{Esc}})}{((N_s - 1) \times P_{\text{Pitch}} + P_{\text{Width}} + O_{\text{Esc}})} \ (4.39)$$
4.1.4.3. Partial Array with Empty Center for Staggered Vias

Area Arrays with depopulated center are more common and can be divided into two types: with an empty center or having a small array of bumps in the empty center. For both types the center area is parameterized by the missing rows in the large \( N_{\text{LF}} \) and the small \( N_{\text{SF}} \) direction. Thus the number of missing bumps including a possible center array is \( N_{\text{LF}} \times N_{\text{SF}} \). We first consider an empty center filled of staggered vias as shown in figure 4.8. To escape as many vias as possible into an empty center of area pads we do not use a fixed via grid but place them as densely as possible. First the model for a square empty center of \( N_{\text{E}}^2 \) missing bumps is explained. Here four vias can be placed in the middle with minimal distances as shown in figure 4.8 because no line has to pass between. Thus, the number of vias in the first row \( V_0 \) is two and the number of vias to escape from inside this first row \( (E_{-1}) \) is 0. For the other rows, \( E_n \) can be calculated with (4.40) and \( V_n \) is calculated as follows:

Every row is at least two times the minimum via grid and two times the line pitch wider than the preceding one. The number of its vias depends on this width, the number of lines to escape \( (E_{n-1}) \) from the predecessors and the via pitch (4.41):

\[
\begin{align*}
V_0 &= 2 \\ E_{-1} &= 0 \\ E_n &= E_{n-1} + V_n - 1 \\ a &= (E_{n-2} + 2 - E_{n-1})(L_{\text{Width}} + L_{\text{Space}}) + L_{\text{Space}} \\ V_n - \text{staggered} &\geq \frac{(V_{n-1} + 2)(D_{\text{ViaLand}} + L_{\text{Space}}) + a}{D_{\text{ViaLand}} + L_{\text{Space}}} \\
\end{align*}
\]

The width of a row \( (W_n) \) is then calculated with (4.42). This width has to be smaller than the center area minus two times the space for one line to pass between the bumps and the vias (4.43). As the via rows are simultaneously added on all four sides, the number of escaped bumps \( (E_{\text{tot}}) \) is four times \( E_n \) (4.44):

\[
\begin{align*}
W_n &= V_n \times (D_{\text{ViaLand}} + L_{\text{Space}}) + L_{\text{Space}} \\ &+ E_{n-1} \times (L_{\text{Width}} + L_{\text{Space}}) \\ W_n &\leq N_{\text{E}} \times P_{\text{Pitch}} - P_{\text{Width}} - 2 \times (L_{\text{Width}} + L_{\text{Space}}) \\ E_{\text{tot}} &= 4 \times E_n
\end{align*}
\]
For a rectangular empty center the missing bumps are parameterized as $N_{IE} + N_{sE}$. Here, the number of starting vias depends on the ratio of $N_{IE}$ and $N_{sE}$. The minimum for $V_{10}$ is one but when $N_{IE}$ is not significantly larger than $N_{sE}$ a minimum of two should be used. Now $V_{10}$ is calculated with (4.45).

$$V_{10} \geq \frac{V_{30} \cdot N_{IE}}{N_{sE}} \quad (4.45)$$

If the center is filled in one direction, some space may be left for vias in the other. In this case, the number of vias ($V_{n_2}$) is calculated in (4.46) as the maximum width is now fixed as the empty area width:

$$V_{n_2} \leq \frac{N_{IE} \cdot P_{pitch} - P_{width} - (E_{n-1} + 2)(L_{width} + L_{space})}{D_{V_{land}} + L_{space}} \quad (4.46)$$

When the number of vias that can be placed in the center ($E_{tot}$) is calculated, it has to be checked if the same number of bumps can be escaped to these vias. Equation (4.47) shows that the maximum is given by the number of rows that can be escaped on the top ($l_{top}$) minus four times $l_{top}^2 - 1$ bumps that can not escape because they are situated in the corners. The number of layers used for escaping is then calculated by using the full array equation (4.37) and enlarging $C_P$ by $E_{tot}$ plus the number of depopulated bumps $N_{IE} \cdot N_{sE}$. Thus (4.37) becomes (4.48):

$$a = (N_{IE} + 2 \cdot l_{top})(N_{IE} + 2 \cdot l_{top}) - N_{IE} \cdot N_{sE} - 4 \cdot (l_{top}^2 - 1)$$

$$E_{tot} - eff = \min(E_{tot}, a) \quad (4.47)$$

$$N_{Layer} = n(C_P + C_P + E_{tot} - eff + N_{IE} \cdot N_{sE}) + 2 \quad (4.48)$$

The presented equations were for staggered vias. For through hole vias equation (4.41), (4.42) and (4.46) have to be adapted as described below.

### 4.1.4.4. Partial Array with Empty Center for Through Hole Vias

The model for through hole vias in the empty center is the same as for staggered vias. However, through hole vias would block the whole area below a component if not spaced properly, a larger via pitch is used to allow $L_{ESV_{vias}} > 1$ lines to pass between them. To do so, (4.41) becomes (4.49), (4.42) (4.50) and (4.46) (4.51).

$$i = L_{ESV_{vias}}$$

$$L_{Pitch} = L_{Space} + L_{Width}$$

$$a = \max(E_{n-2} - (V_{n-1} - 1) \cdot i; 0)$$

$$b = \max(E_{n-1} - (V_{n-1} - 1) \cdot i; 0)$$

$$c = D_{V_{land}} + i \cdot L_{Width} + (i + 1) \cdot L_{Space}$$

$$V_{n} \geq \frac{(V_{n-1} + 1) \cdot c \cdot L_{Space} \cdot (i \cdot a \cdot b) \cdot L_{Pitch}}{D_{V_{land}} + i \cdot L_{Width} + (i + 1) \cdot L_{Space}} \quad (4.49)$$

$$d = \max(E_{n-1} - (V_{n-1} - 1) \cdot i; 0)$$

$$W_{n} = V_{n} \cdot c + L_{Space} + d \cdot L_{Pitch} \quad (4.50)$$

$$V_{n_2} \geq \frac{N_{IE} \cdot P_{pitch} - P_{width} - (2 \cdot b + b) \cdot L_{Pitch}}{D_{V_{land}} + i \cdot L_{Width} + (i + 1) \cdot L_{Space}} \quad (4.51)$$

The empty center can also have a small array as described below.
4.1.4.5. Partial Array with Array in Empty Center

If the depopulated center is filled with an array of \( N_{if} \) bumps, their escaping overhead \( (O_{esc}) \) is calculated using the same equations as for full array. Equation (4.52) then calculates the necessary area \( (A_{center}) \). If it is larger than the total inner area including the center bumps \( (A_{empty}) \) as calculated in 4.53, \( l_{top} \) has to be reduced for escaping calculation. This costs additional layers.

\[
A_{center} = ((N_{if} - 1) \times P_{pitch} + O_{esc}) \\
A_{empty} = N_{if} \times (P_{pitch})^2 \times N_{se}
\]

(4.52)

(4.53)

If there is some space left after escaping the center array, vias can be placed as described for an empty center. The number of vias to start with \( (V_{0}, V_{s0}) \) is calculated from the width of the center area already used with equations (4.54). The number of layers used is then calculated with equation (4.48) plus the number of layers used for the center array escaping \( (n_{center} + 1) \) as shown in equation (4.56).

\[
a = (N_{if} - 1) \times P_{pitch} + P_{width} + O_{esc} + 2 \times (D_{Via,land} + L_{space})
\]

\[
V_{0} - stagger = \frac{a}{D_{Via,land} + L_{space}}
\]

\[
V_{0} - through = \frac{a}{D_{Via,land} + L_{esc} + L_{pitch} + L_{space}}
\]

\[
N_{layer} = n(C_{p} = C_{p} + E_{int} + N_{if} \times N_{se}) + 2 + n_{center} + 1
\]

(4.54)

(4.55)

(4.56)

4.1.5. Wirebonding

Wirebonding adds some overhead because substrate bond pads especially on laminates have to be larger than on the ICs. The pad size is adapted to allow for misalignment, the softer substrate and the larger footprint for the second bond for ball-wedge bonding. In the following equation the number of necessary wirebond rows and their spacing is calculated before the footprint estimation is done.

First orthogonal wire bonding is introduced, which is best suited for wedge-wedge bonding. Then a model for any angle wire bonding is presented which calculates wire bonding that spreads the bond pad rows.

4.1.5.1. Orthogonal Bonding/Wedge-Wedge Bonding

For wedge-wedge bonding orthogonal wire-bonding is preferred, because turning the bond head costs time. Therefore, the minimum number of rows depends on the minimum on-chip pad pitch \( (C_{pp}) \) versus the minimum feasible bondpad pitch on the substrate \( (B_{width} + L_{space}) \) as shown in formula 4.57. This is the worst case because the mean on-chip pitch is larger and slight spreading can be used (see section 4.1.5.2). But often there are depopulated areas, so a mean pitch calculation is overoptimistic.

\[
N_{row} \geq (B_{width} + L_{space})/C_{pp}
\]

(4.57)

The spacing between two wire bond rows is calculated from the via land \( (D_{Via,land}) \), the line width and spacing \( (L_{width}, L_{space}) \) and the bondpad pitch. It mainly depends on the number of vias required to reach the next layer without blocking it completely. The number of vias in one
line, required to escape from one wire bond, \((k)\) row can be estimated with equation (4.4) or (4.5) as illustrated in figure 4.3 using:

\[
Pitch = \max(N_{Row} \times C_{pp}; BP_{width} + L_{space})
\]  

(4.58)

The overhead is then calculated with equation 4.59 including the minimum bondpad to bondpad spacing \((B_{space})\) of adjacent ICs. \(d_{Esc}\) is calculated from (4.6) or (4.7). If the bondpad rows are escaped to both sides, it has to be considered that \(k2 > k1\) results in shorter bond wires for the outermost row. But the footprint is larger because the escaping distance from the bondpads in the direction of the chip \((d_{ESC2})\) is smaller than \(d_{ESC}\) and is calculated in (4.11). The number of layers blocked for escaping is the same as the number of bondpad rows (4.60).

\[
\begin{align*}
\alpha &= \text{if}(k1 > 0; D_{ViaLand} + L_{space}) + (k1 - 1) \times d_{ESC2}; 0) \\
\beta &= \text{if}(k2 > 0; D_{ViaLand} + L_{space}) + (k2 - 1) \times d_{ESC}; 0) \\
C2P &= \max(C_{height}; C2P_{min} + \alpha) \\
O_{ESC} &= C2P + BP_{length} + (N_{Row} - 1) \times b \times \max(\frac{B_{space}}{2}; D_{ViaLand} + 2 \times L_{space} + k2 \times d_{ESC}) \\
N_{Layer} &= N_{Row} \times \left(1 + (BP_{length} + L_{space} + \alpha) \right)
\end{align*}
\]  

(4.59)

(4.60)

Assembly companies often specify the minimal \((L_{Min})\) and maximal \((L_{Max})\) horizontal wire bond length. The minimum is calculated from the chip to substrate bond pad spacing \((C2P)\), the bond pad length and the chip pad to chip border distance \((P2C)\) (4.61). The maximum length is the escaping overhead minus the escaping of the last row (4.63). For the inductance estimation the real length \((L_{Min-eff}, L_{Max-eff})\) is calculated as specified in the simple model of the JEDEC standard [59]. It depends on the horizontal distance, the wire loop height \((H_{Loop})\), the chip height \((C_{height})\) and the height of the chip attachment \((C_A_{Height})\) (4.62,4.64):

\[
L_{Min} = C2P + \frac{BP_{length}}{2} + P2C
\]  

(4.61)

\[
L_{Min-eff} = H_{Loop} + L_{Min} \times \frac{8}{7} + \sqrt{\left(\frac{7 \times L_{Min}}{8}\right)^2 + (C_{Height} + C_A_{Height})^2}
\]  

(4.62)

\[
L_{Max} = \frac{O_{ESC}}{2} + P2C
\]  

(4.63)

\[
L_{Max-eff} = H_{Loop} + L_{Max} \times \frac{8}{7} + \sqrt{\left(\frac{7 \times L_{Max}}{8}\right)^2 + (C_{Height} + C_A_{Height})^2}
\]  

(4.64)

4.1.5.2. Any Angle Bonding/Ball-Wedge Bonding

For ball-wedge bonding any angle bonding can be used because the bond capilar does not have to turn and there is no stub at the first bond that might short circuit to its neighbour. But spreading
can also be used for wedge-wedge bonding. Here, the maximum on-chip angle on the bond pads ($\beta_{CP}$) is smaller than 45 degree. Another parameter that influences the spreading is the maximum bond wire to bond pad misalignment angle ($\Delta \beta_{BP}$). This parameter should be specified by the assembly partner. If the misalignment is smaller than $\Delta \beta_{BP}$, the bond pads remain orthogonal or are parallel to the preceding ones. The minimum number of wire bond rows can first be estimated with equation (4.65). It depends on the distance from the chip to the first bond pad row ($C2P$) and the number of pads per side of a chip ($CPPS$). All parameters calculated below are first order estimations. They are checked later because the number of rows ($N_{Row}$) might be increased in order to reduce $\beta_{CP}$.

\[
Pitch_1 = \max(C_{pp}, BP_{width} + L_{space})
\]
\[
e = \epsilon f(k1 > 0; D_{via} + L_{space} + (k1 - 1) * d_{ES,2}; 0)
\]
\[
C2P = \max(C_{height}, C2P_{min} + e)
\]
\[
CPPS = C_N/4 \text{ or real number}
\]
\[
N_{Row} \geq CPPS \times \frac{BP_{width} + L_{space}}{C_{length/width} + 2 \times C2P - \sqrt{2}L_{space}}
\] (4.65)

![Diagram showing spreading for wirebonding with and without aligning](image)

As shown in figure 4.9, the pads of a wire bond row have all the same distance from the IC. This is $C2P$ for the first row. To calculate the angle of the bonds we start in the center of the first bond row. Its angle ($\beta_{10}$) is always 0 degree. As next step, the angle of the two neighboring pads (left and right) is calculated simultaneously. Their angles ($\beta_{11}$) depend on the horizontal misalignment between the chip pad and the substrate bondpad ($\Delta X_{11}$). Equation (4.66) calculates their and the other angles of the first wire bond row ($\beta_{1eff}$). From these values $\beta_{1n}$ is derived, which is used for the further calculations. Angle $\beta_{1n}$ is the same as $\beta_{1eff}$ or is made as small as the
preceding one \((\beta_{1_{n-1}})\) if the resulting wire to bondpad misalignment is small enough i.e. smaller than \(\Delta \beta_{BP}\):

\[
\beta_{eff_n} = 90 - \arctan \frac{P2C + C2P}{\Delta X_n} - \arcsin \frac{BP_{width}}{2\sqrt{\Delta X_n^2 + (P2C + C2P)^2}}
\]

\[
\beta_{1_n} = \beta_{1_{n-1}} \text{ if } \beta_{eff_n} < \beta_{1_{n-1}} + \Delta \beta_{BP} \text{ else } \beta_{eff_n}
\]

\[
\beta_{1_0} = 0
\]

The horizontal mismatch \((\Delta X_n)\) would depend on the horizontal distance from bondpad to bondpad \((d_n)\) measured from corner to corner situated on the minimum distance line \((C2P)\) (4.67). But \(d_n\) cannot be calculated before knowing \(\beta_n\) which is calculated from \(\Delta X_n\). Thus, \(d_n\) is estimated as being the same as the previous one \((d_{n-1})\). This approach results in a negligible mismatch for \(\beta_{1_n}\) smaller than one degree.

\[
e = N_{Row} \cdot C_{pp} \text{ or real distance}
\]

\[
\Delta X_n = \max(\Delta X_{eff_{n-1}} + d_{n-1} - c; BP_{width}/2)
\]

\[
\Delta X_0 = \frac{BP_{width}}{2}
\]

\[
\Delta X_{eff_0} = \Delta X_0
\]

\[
d_{1_0} = \frac{BP_{width} + L_{space or real distance}}{2}
\]

After estimating the horizontal mismatch \(\Delta X_n\) in 4.67 the angle \(\beta_n\) is known. Thus, the distance \(d_{1_n}\) is calculated with equation (4.68). With this distance, the effective horizontal mismatch \(\Delta X_{eff_n}\) is calculated in equation (4.69):

\[
d_{1_n} = \max(BP_{width} \cdot \cos(\beta_{1_n})
\]

\[
+ L_{space} \cdot \cos(\beta_{1_{n-1}}) + (BP_{width} \cdot \sin(\beta_{1_n})
\]

\[
\cdot L_{space} \cdot \sin(\beta_{1_{n-1}})) \cdot \tan(\beta_{1_{n-1}}); d_{1_0})
\]

\[
\Delta X_{eff_{n-1}} = \max(\Delta X_{eff_{n-2}} + d_{n-1} - a; BP_{width}/2)
\]

The number of pads \((N_{Pads})\) bonded with \(N_{Row}\) wire bond rows and \(n\) pads to both sides is calculated in (4.70). Equation (4.71) uses this relation to calculate the number of pads required on both sides \(n_{max}\). Equation (4.72) then calculates the width of the first wire bond row \((W1_{Row})\). Afterwards, the feasibility of the estimated number of rows \((N_{Row})\) is checked with (4.73):

\[
N_{Pads} = n \cdot 2 \cdot N_{Row} + 1
\]

\[
n_{max} \geq \frac{C_{pp} S - 1}{2 \cdot N_{Row}}
\]

\[
W1_{Row} = 2 \cdot (n_{max} \cdot N_{Row} \cdot C_{pp or real distance} + \Delta X_{eff_{n_{max}}})
\]

\[
c = C2P - \sqrt{2}L_{space} - BP_{length} \cdot \cos(\beta_{n_{max}})
\]

\[
Feasible\ if \ W1_{Row} \leq C_{length/(a,c)} + 2 \cdot c
\]

\[
and \beta_{eff_{n_{max}}} \leq \beta_{BP}
\]

For the outer wire bond rows the angle and the position of a bondpad depend on the first row. The distance from bondpad center to bondpad center of two bondpads in the first rows is divided by the number of wire bond rows. At these positions the corresponding wires cross equidistantly. If
the bondpad configuration on the chip is not too irregular, this strategy results in a very regular bonding arrangement with increasing angle \( \beta \) starting from the center of the chip bond pad rows. Equation (4.74) calculates the horizontal angle \( \beta \) starting from the center of the on-chip bondpad and the crossing point with the first row (\( \Delta X_{i_n} \)). Here the \( i \) signifies the \( i \)th row. The number \( n \) starts for the outer rows at one. With \( \Delta X_{i_n} \) the corresponding angle of the bondpad (\( \beta_{i_n} \)) can be calculated (4.75):

\[
\begin{align*}
b &= \frac{BP_{\text{width}}}{2} \times \cos \beta_{1_n} - \frac{BP_{\text{length}}}{2} \times \sin (\beta_{1_n}) \\
a &= \frac{BP_{\text{width}}}{2} \times \cos \beta_{1_n-1} - \frac{BP_{\text{length}}}{2} \times \sin (\beta_{1_n-1}) \\
\Delta X_{i_n} &= \Delta X_{1_n} - ((i - 1) \times C_{pp} \text{ or real distance}) \\
&+ \frac{i - 1}{N_{\text{Row}}} (d_{1_n} + a - b) - a \\
\beta_{i_n} &= \arctan \left( \frac{\Delta X_{i_n}}{BP_{\text{length}}} \right). 
\end{align*}
\]

In order to calculate the position of the pads, the distance between two wire bond rows (\( B2B_i \)) has to be calculated. This distance between the \( i \)-1th and the \( i \)th row depends on the escaping. This escaping parameters \( k1, 2 \) and \( d_{E_{\text{esc},i}} \) are calculated with the standard equations 4.5-4.11 with a different \( \text{Pitch} \) for the rows. As spreading is used, the pitch is larger than the minimum wire bond pitch (\( BP_{\text{width}} + L_{\text{space}} \)). Taking the second smallest distance gives a good approximation. It is measured as outer distance between the bondpad 1 and 2 starting from the center of the die of the \( i \)th row (4.76). Equation 4.77 calculates \( B2B_i \) from the escaping parameters. Here the pitch of the preceding line (\( i - 1 \)) is already known but the pitch of the \( i \)th line can not be calculated before \( d_{i_n} \) is known. Thus \( k1 \) should be set to zero or one. Knowing \( B2B_i \) allows calculating \( d_{i_n} \) (4.78). Afterwards \( B2B_i \) and \( d_{i_n} \) can be corrected iteratively:

\[
\begin{align*}
Pitch_i &= d_{i_n} + BP_{\text{length}} \times (\cos (\beta_{i_n}) - \cos (\beta_{1_n})) \\
B2B_i &= BP_{\text{length}} \times i (k2_{i-1}) > 0; D_{\text{via,Laod}} + L_{\text{space}}; 0) \\
&+ (k2_{i-1} - 1) \times d_{E_{\text{esc},i-1}} + i (k1_i > 0; D_{\text{via,Laod}} + L_{\text{space}}; 0) \\
&+ (k1_i - 1) \times d_{E_{\text{esc},i}} + L_{\text{space}} \\
d_{i_n} &= (N_{\text{Row}} \times C_{pp} \text{ or real distance}) \times BP_{\text{width}} \times \cos \beta_{1_n-1} + BP_{\text{width}} \times \cos \beta_{1_n} \\
&+ \left[ P2C + C2P + \sum_{i=2}^{i=n} B2B_i \right] \times (\tan (\beta_{i_n}) - \tan (\beta_{i_n-1})) \\
d_{i-1} &= (i \times C_{pp} \text{ or real distance}) \times BP_{\text{width}} \times \frac{1}{2} - \frac{1}{\cos \beta_{1_n}} \\
&+ [P2C + C2P + \sum_{i=2}^{i=n} B2B_i] \times \tan (\beta_{i_n}) 
\end{align*}
\]

The escaping overhead for this spreading model is given in equation (4.79). The local layer congestion is the same as for the orthogonal wire bonding (4.80).

\[
\begin{align*}
\frac{O_{\text{Esc}}}{2} &= C2P + \max \left( \frac{D_{\text{space}}}{2}; D_{\text{via,Laod}} + 2 \times L_{\text{space}} \right) \\
&+ k2 \times N_{\text{Row}} \times d_{E_{\text{esc},i}} \times BP_{\text{length}} + \sum_{i=2}^{i=n} B2B_i \\
N_{\text{Layer}} &= N_{\text{Row}} 
\end{align*}
\]
Chapter 4: Component Footprint and Chip Limited Substrate Size

The minimum horizontal wire bond length \( l_{\text{Min}} \) and the effective wire length \( l_{\text{Min-eff}} \) are calculated as for the orthogonal wire bonding (4.61, 4.62 and 4.64). But the horizontal maximum length \( L_{\text{Max}} \) is different (4.81):

\[
\begin{align*}
  c &= \max \left( \frac{B_{\text{space}}}{2}, D_{\text{Vialand}} + 2 \cdot L_{\text{space}} + (k2_{\text{Hrow}} - 1) \cdot d_{\text{EscHrow}} \right) \\
  l_{\text{Max}} &= \frac{0.5 \cdot \sqrt{b_{\text{Max}}^2 - c^2} + P2C}{\cos \left( \frac{\pi}{3} N_{\text{Hrow max}} \right)}
\end{align*}
\] (4.81)

### 4.1.6. BGA/CSP

The footprint of a BGA or a CSP is calculated with the same equations as flip chip described in subsection 4.1.4. Thus, in this section only custom packaging of a die into an area I/O package is modeled. These are not standard packages. For them, a die is virtually connected with wire bond, flip chip or TAB onto an interposer or rerouted on-chip. We first consider rerouted ICs where the size is the silicon area or marginally larger.

#### 4.1.6.1. CSP using Rerouting

![CSP using Rerouting Diagram](image)

**Figure 4.10:** Rerouted ICs shown as white rectangle. The peripheral pads (black rectangles) are rerouted to area pads (black circles). The gray rectangle is the area allowed for the rerouted pads.

CSPs made by rerouting the ICs are limited to the silicon area. For peripheral pads on-chip, the area for the rerouted pads is even smaller because the new pads might not be placed onto the IC pads. This restriction is modelled as a border \( C2P_{\text{CSP}} \) from the chip edge. This border might even be larger when the rerouting is done with a TAB because space is needed for the inner lead bonding. As rule of thumb the border width \( C2P_{\text{CSP}} \) is five mm. Thus, the gray area in figure 4.10 shows the area that can be used for the area pads. Their maximum pitch is calculated with the option full area array as shown on the right side of figure 4.10. This might not be the best solution because the via lands are usually much larger than the line width and space. The easiest configuration to escape is a two row approach because maximum one line has to pass between the area pads on the substrate. The inner row might even be escaped to the empty area.
However, rerouted ICs are limited to a moderate number of I/Os because for high pin count components the pitch becomes very small due to the limited area. To overcome this limit an interposer can be used as described below.

4.1.6.2. BGA and CSP with Interposer

Area I/O packages with interposer have the advantage that even ICs with a lot of I/Os can be connected at a reasonable pitch. Its size is either defined by the first level interconnection onto the interposer or the number of pins and its configuration. In a first step the size governed by the IC and its interconnection is calculated using the equations in section 4.1. Then suitable pitches and configurations such as full array, empty center or empty center with array in center are evaluated with the equations in subsection 4.1.4. Finally, a pitch and configuration are chosen for different substrate types.

4.2. Footprint Examples

In this section, the equations presented in this chapter are checked and used to calculate the examples. First the footprint for a pentium processor is calculated for several bonding options on all types of substrate. Then area I/O is discussed and finally an example for wire bonding with spreading is presented.

4.2.1. Pentium Processor

For the following examples I present a pentium processor with voltage reduction technology from Intel [60]. This component is available in a ceramic pin grid array (PGA), in a TAB package and as fully tested bare die (smart die). The die measures 9.1 mm × 9.9 mm and has 35 peripheral pads on a 75 μm pitch. This IC is packaged virtually into standard package PGA, in a chip size package (CSP) and mounted using TAB, wire bond and flip chip (FC) as first level interconnect.

These packages are placed on all generic types of substrate such as PCB, SBU, conventional thick film (MCM-C) and thin film (MCM-D). Their wiring pitch (P_wired) range from 65 μm to 510 μm. Their line width (L_width), line space (L_space), via land diameter (D_via) and their minimum bond pad width (B_width) are summarized in table 4.2.

<table>
<thead>
<tr>
<th>Table 4.1: Processor footprint for different packages/first level interconnect and substrates</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bonded Method</strong></td>
</tr>
<tr>
<td><strong>Substrate</strong></td>
</tr>
<tr>
<td><strong>Package Size [mm²]</strong></td>
</tr>
<tr>
<td><strong>Footprint Overhead [mm²]</strong></td>
</tr>
<tr>
<td><strong>Keepout Size [mm²]</strong></td>
</tr>
<tr>
<td><strong>Num. of Layers for Escap- ing</strong></td>
</tr>
<tr>
<td><strong>4.2.1.1. PGA</strong></td>
</tr>
</tbody>
</table>

The processor is available in a 296 pin ceramic pin grid array measuring 50*50 mm. This package is easy to fanout even on a standard PCB. But its package size as well as the package parasitics are major drawbacks. In fact, the PGA consumes more than double the area as the next smaller first level interconnect TAB shown in table 4.1 and figure 4.11.
4.2.1.2. TAB

Besides its large size due to the high number of pins the TAB mounting, figure 4.11(a), suffers from the large escaping overhead needed on substrates with lower interconnect density. Even when some pins escape under the body, the escaping overhead to be added to the package size is more than 8 mm per side. Furthermore, the assembly of TAB is not standard as special equipment has to be used.

4.2.1.3. BGA

BGA packages are much smaller. When packaging the processor virtually in a BGA, we found that the package size would be 420 mm² at 1 mm ball pitch. This I/O defined size is only slightly larger than the size needed to wire bond the die onto the BGA interposer. When placing this package onto a PCB, the footprint is the size of the pads enlarged by the fanout overhead. The resulting footprint of 543 mm² is caused by the required three via rows. The package placed on an SBU results in an escaping size being smaller than the footprint because the 2 mm space required between adjacent components consumes more area than escaping.

4.2.1.4. Wire Bond

Wire bonding is the most mature and most widely used first level interconnect. It allows to match the die bond pad pitch to the much larger substrate bondpad pitch as long as the bond wires do not become too long. This substrate pitch is highly dependent on the technology as shown in table 4.2. Whereas the pin pitch ratio (on-chip vs. substrate) defines the number of wire bond rows, the design rules dominate the required space between two rows to allow escaping to an inner layer. Thus, the fanout overhead on an SBU is four times larger than on an MCM-D. As shown in table 4.3 this 8.7 mm overhead is due to three bond rows needed instead of two for thin film and the much larger spacings between caused by the vias. Thus, the footprint on an SBU is larger than three times the die size whereas for an MCM-D it is only 50% overhead to the die as shown in figure 4.11(c) and 4.11(d). These figures are pictures of existing modules [44], [61].

4.2.1.5. CSP and FC

The smallest possible footprint can be obtained with flip chip (FC) mounting. Our example is difﬁcult to bond directly as the bond pad pitch is too small. This can be overcome by adding a thin film layer onto the IC to re-route the peripheral pads to area interconnections, thus making a chip size package (CSP). This CSP has a much larger pitch of 430 μm. It is not possible to escape this CSP on a PCB but the footprint is the same on an SBU and MCM-D because it is dominated by the minimum spacing between the chips (1mm). Obviously more layers are needed on an SBU substrate.
The same footprint results for FC without re-routing. But it can only be bonded onto an MCM-D because of the very tight line pitch needed.

Figure 4.11: Footprint comparison for a 9.1 * 9.9 mm mobile pentium processor die (white). The escaping size is the dashed box and the footprint size the grey box.

4.2.1.6. Calculation Results

Table 4.3 specifies some more input data and shows the intermediate data for the calculation of the sizes in table 4.1. This data as well as the resulting sizes were calculated using the models presented in the previous section. It should be noted that for the TAB calculation k would be 11 if not $k_1 = 5$ out of them were routed to the inside.
4.2.2. Area I/O

To demonstrate the Area I/O model I consider an other IC than before. This IC features $C_N = 100$ pins. $C_N$ includes $C_P = 10$ power pins. This IC will be packaged into a QFP and BGAs with 1, 0.75 and 0.5 mm pad pitch. The packages are then mounted on a standard PCB as well as on sequential build-up layers (SBU). The substrate design rules are summarized in Table 4.2. In these examples the minimum distance between adjacent components is 2 mm. The summary of the comparison is presented in Table 4.4 and the results are discussed in the subsections below.

### Table 4.3: Footprint calculations

<table>
<thead>
<tr>
<th>Bonding Method</th>
<th>PGA</th>
<th>TAB</th>
<th>BGA A</th>
<th>BGA B</th>
<th>BGA C</th>
<th>BGA C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pad Pitch [mil]</td>
<td>2540</td>
<td>1000</td>
<td>1500</td>
<td>150</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>Pad width [mil]</td>
<td>1000</td>
<td>1000</td>
<td>630</td>
<td>700</td>
<td>700</td>
<td>700</td>
</tr>
<tr>
<td>Keypin [mil]</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Stagger [mil]</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Eta</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$e$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$d_{g-k}$ [mil]</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

4.2.2.1. QFP

The QFP is easy to mount because its footprint is dominated by component size and the necessary keepout as shown in figure 4.1(a). The escaping can then be done on one or two layers. But it has by far the largest footprint of all examples.

4.2.2.2. BGA A (1 mm Pad Pitch)

A full array with 1 mm pad pitch is less than half the size of a QFP. And it can be routed on two layers on a standard PCB. However, due to the concentration of the I/Os on a small pitch and its large vias, a large escaping overhead of about 6 mm has to be paid. Thus, the footprint is nearly as large as the QFP package size. As only one trace can pass between the vias, only two rows ($t_{top} = 2$) are escaped on the top layer. The rest can escape on a second layer which can be reached because the via lands fit between the balls. To reduce the footprint to the keepout size one build-up layer is sufficient.
By reducing the pad pitch the package can be made smaller (BGA B). But a full area array is not routable on a PCB because no vias can be placed between the pads. At the cost of two SBU's this package can be used with a footprint dominated by the keepout.

Another approach is to use a BGA with an empty center area. On a PCB no trace can pass between these balls. Therefore, only a ring of two rows can be used. The minimum size that is able to provide 100 balls is a BGA of 12x13 (N_s * N_t) balls with an empty center of 10x11 (N_{sE} * N_{tE}). Equation (4.44) shows that all inner balls can be escaped into the center. As shown in table 4.1 this package BGA B2 has about the same footprint as package A with one SBU. Thus, package B2 is preferred.

A full area array with a 0.5 mm pad pitch (BGA C) is still routable on SBU because vias can be put between the balls. It is even possible to put them into the pads. The very small package footprint is dominated by the keepout area. In fact the footprint is only a fourth of the BGA A mounted on PCB. But the escaping requires three SBU layers which rends the substrate expensive.

To reduce the area usage the center is depopulated in version C2. It needs the same depopulated center as B2 as no SBU trace can pass between the balls. This depopulation saves one SBU layer, but the package has about the same footprint as B mounted on SBU. Therefore, B is preferred due to the better assembly yield.

The above packages were calculated from the parameters in table 4.4. As background information the intermediate results for the above packages are summarized in table 4.5. It can be seen that k for the BGA A on PCB is quite high which causes the large escaping overhead. For the BGAs with the depopulated center 46 vias have to be placed in the empty area. For the B2 three rows with 2,5,6 vias are used and on one side another 6 vias are added. Thus, the required number just fits into the empty area. In the smaller empty center of C2 more vias could be placed due to the smaller via lands and line pitches. But as no line can pass between the pads, only the innermost row with its 46 pads can be reached.

<table>
<thead>
<tr>
<th>Substrate</th>
<th>QFP PCB</th>
<th>BGA A PCB</th>
<th>BGA C PCB</th>
<th>BGA A SBU</th>
<th>BGA B SBU</th>
<th>BGA C SBU</th>
<th>BGA C2 SBU</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_{tot (effective)}</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>46(46)</td>
<td>n/a</td>
<td>64(46)</td>
</tr>
<tr>
<td>l_{op-eff}</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>l_{in}</td>
<td>n/a</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>k</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>d_{G-sc} [\mu m]</td>
<td>n/a</td>
<td>685</td>
<td>172</td>
<td>n/a</td>
<td>659</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>
4.2.3. Wire Bond Spreading

Figure 4.9 shows a wire bonded IC measuring 3.5 * 3.5 mm with \( C_N = 100 \) pads. The IC's minimum pad pitch \( (C_{PP}) \) is 100 \( \mu m \) with a distance \( P2C = 150 \mu m \) from the die edge. It is bonded onto \( BP_{width} = 150 \mu m \) wide bond pads with a minimum distance of \( I_{space} = 75 \mu m \) using two wire bond rows. On the left side of figure 4.9 a wire to bond pad misalignment of \( \Delta \beta_{BP} = 10 \) degree is allowed. As the misalignment is smaller no alignment is necessary. On the right side the wire and the bond pad are always aligned. For this configuration the calculated values such as bond pad angle \( (\beta 1) \) and the distance from pad to pad \( d1 \) for the first row can be found in table 4.6. Here, 25 pins have to be bonded on each side of the IC resulting in a bond row width \( W1_{Row} \) of 2867 \( \mu m \) and a maximum angle of 13 degree.

![Table 4.6: Values for figure 4.9](image)

<table>
<thead>
<tr>
<th>N of Pins</th>
<th>( \Delta X1 ) ( \mu m )</th>
<th>( \beta 1 )</th>
<th>( d1 ) ( \mu m )</th>
<th>( \Delta X1_{eff} ) ( \mu m )</th>
<th>( W1_{Row} ) ( \mu m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>75</td>
<td>0</td>
<td>225</td>
<td>75</td>
<td>150</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>2</td>
<td>225</td>
<td>100</td>
<td>600</td>
</tr>
<tr>
<td>9</td>
<td>125</td>
<td>4</td>
<td>225</td>
<td>125</td>
<td>1050</td>
</tr>
<tr>
<td>13</td>
<td>130</td>
<td>7</td>
<td>226</td>
<td>150</td>
<td>1501</td>
</tr>
<tr>
<td>17</td>
<td>176</td>
<td>9</td>
<td>226</td>
<td>177</td>
<td>1954</td>
</tr>
<tr>
<td>21</td>
<td>203</td>
<td>11</td>
<td>228</td>
<td>204</td>
<td>2409</td>
</tr>
<tr>
<td>25</td>
<td>232</td>
<td>13</td>
<td>229</td>
<td>233</td>
<td>2867</td>
</tr>
</tbody>
</table>

4.3. Chip Limited Substrate Size Estimation

The chip limited size of a module is the minimal size. It can only be realised if the wiring capacity is high enough to route on this size on a reasonable number of layers. The component limited size is the sum of the area required to place the components \( (A_{Comp}) \) and the area for the connection of the module to the next level \( (A_{Inter}) \) (4.82). The module size can be larger than the substrate \( (A_{ChipLimited}) \) as the packaging of the substrate often needs additional area. Or the package can even define the substrate size.

\[
A_{ChipLimited} = A_{Comp} + A_{Inter} \tag{4.82}
\]

The basis for the estimation of the component area are their footprints. The simplest model is then the sum of these areas. This strategy normally is an underestimation as shown in section 3.6 because area is lost by non-ideal chip form factors. A placement gives the most accurate results because not only the form factors but also the interconnectivity are considered. Therefore, other algorithms have to be investigated. The following discussion considers placement only on one side. But the strategies can easily be adapted for double sided placement.

4.3.1. Sum of Areas

Sum of area is a very fast size estimation. However, it provides only good results if more than 50 % of the component area are very small components such as SMT passives (figure 4.12). Otherwise there is a lot of area lost between the components as shown in figure 3.7(a). Thus, the simplest correction is to multiply the sum with a factor higher than 1. Good results were obtained by adding 10 %.
4.3. Chip Limited Substrate Size Estimation

Figure 4.12: Example for a module whose size can be estimated with 'Sum of Areas' because its size is defined by the many small passives.

4.3.2. Simple Placement

Figure 4.13: Simple Placement

Simple placement[62] is an algorithm that places a list of components as densely as possible without considering its interconnections. It starts with the largest component and places the subsequent ones in a spiral form as shown in figure 4.13:

1. Sort out external connectors
2. Place the component with the largest footprint.
3. Place the next smaller component on the right and on the bottom.
4. Rotate the component by 90 degree and place it on the right and on the bottom.
5. Keep the position which shows smallest overall dimensions.
6. Move this component to the bottom if placed right or to the left if placed on the bottom.
7. Place the next smaller component next to the previous one if it still touches the inner components on the same side as the previous component. If not, turn around the corner and fit it in there to form a counterclockwise spiral.

8. Turn the component by 90 degree and keep the position which has smaller overall dimensions.

9. If the next smaller component is not a small passive continue at 7.

10. Calculate the outline and add to the area the sum of area of the remaining passives.

After performing this task the component area ($A_{Comp}$) is estimated. The external connectors or space for second level interconnect is added in subsection 4.3.4. The above strategy forms a quadratic size. A rectangle is reached by adding a boundary condition in one direction. When it is reached the components are only added in the other direction.

4.3.3. Autoplacement

The simple placement provides good results of a first order estimation. It does not consider the interconnections between the ICs. When a block diagram including the number of interconnections between the blocks is available an autoplacement can be done. As first step a constructive placement is done, where one component after the other is placed fitting to the already placed ones. This initial placement could then be improved by iterations [63]. For this application iterative improvements are not yet possible and necessary as only the minimum size is required and because the build up is not yet defined. Thus, the parameters to calculate the time of flight are not available. After an initial placement a worst case interconnection delay can be done.

The simplest constructive placement algorithms are cluster growth algorithms [64]. They work on a partially placed design. First an unplaced component is selected and then it is placed on the actually best position found [65]. Based on this strategy I propose the following algorithm:

1. Place the component with the largest footprint in the center of the substrate.

2. Select the unplaced component which has the most interconnections to the placed ones or the tightest time constraints.

3. Derive the component position on all four sides (left, right, top or bottom) which uses the minimum area.

4. Turn the component by 90 degree and keep the rotation which has smaller overall dimensions.

5. Calculate the product number of interconnection multiplied by their length in x- and y-direction.

6. Place the component at the minimum product.

7. Continue at 2 if not all components are placed.

8. Optimize for rectangular shape by moving the outermost components clockwise.

9. Calculate the manhattan distances if speed limits are defined.

10. Calculate bounding box
4.3.4. Chip Limited Size

After the calculation of the component area, the area for the connectors to the outside or any other second level interconnect has to be added. For connectors it is their footprint, for wire bond and TAB it is the pad area and for BGA nothing has to be added. The second level interconnect can require more than just the pad area because the pad pitch and the number of I/Os can define the size. Therefore, the resulting pitch for wire bond, TAB and BGA has to be checked.

After this considerations the minimal size for the substrate ($A_{\text{ChipLimited}}$) is known and the routing thermal behaviour as well as the routing can be analyzed.

4.4. Summary and Conclusion

In this chapter, the effective component size, the so called footprint, was defined and models to calculate them were provided. The correctness and the use of them where demonstrated on examples. As all the models are purely geometrical they remain valid even when the NTRS roadmap [66] foresees smaller and smaller pitches.

Based on these footprints the minimum substrate size (chip limited size) is estimated preferably by a rough placement. The routing of this minimum size or the necessary spreading is then calculated in the next chapter.
Chapter 4: Component Footprint and Chip Limited Substrate Size
Routing Limited Substrate Size

In section 4.1 the layer usage for the escaping is calculated. But often the routing which interconnects the components defines layer or area usage. The routing resources available depend on the substrate area, the number of layers, the substrate design rules and the areas blocked for escaping or thermal purposes. Thus, the aim of the models in this section is to calculate the number of layers versus the substrate size, dependent on the design rules.

The required resources depend on the number of interconnections and their length. When the substrate is enlarged to improve the routing resources, the interconnections get longer. As the routing resources grow quadratically and the interconnect length only linearly, it is still a good strategy to reduce the layer count.

In the first section enhancements of Seraphim’s method are presented. It is a first order estimation and works with very little data. Therefore, it can only estimate the routing globally. As soon as more information is available an estimation can include possible local congestions, and instead of estimating the wiring length with Seraphim, it is calculated. To do so, a new routing estimation was developed based on global routing as presented in the other sections of this chapter. The routing of a simplified netlist is performed on a channel graph. The channels between the components are enlarged to provide the routing resources required.

After the calculations in this section the possible substrate selection, their layer counts and their sizes are known.

5.1. First Order Routing Estimation

The first order model is used when only minimum information of a system is available. Thus, these models can be used when only a list of parts and number of I/Os and a placement are available. Here, the routing is only estimated globally because local congestions are completely unknown. A good approach is the estimation of Seraphim. Seraphim only requires the average chip pitch, the number of I/Os and the wiring pitch of the substrate. To adapt the chip pitch when the substrate is spread, the original equation was enhanced as described in the following subsection.

5.1.1. Enhancement of Seraphim’s Estimation

As shown in section 3.9 Seraphim’s method [17] is a simple approach to estimate the required wiring length \( L \) without having any netlist. Using the estimated wiring length, the number of
necessary routing layers can be calculated with (5.1). However, to draw an accurate graph of the number of layers versus the substrate size (figure 3.11), the estimated wiring length has to be lengthened because the substrate area is enlarged (5.2). Here, \( L_0 \) is the calculated wiring length on the original area \( A_w0 \). Thus, the wiring length is multiplied by the substrate edge elongation whereas the area is growing by the power of two as shown in equation (5.3), which is the combination of (5.1) and (5.2):

\[
N_{Layer} = \frac{L * P_w}{A_w} \quad (5.1)
\]

\[
L = L_0 \frac{\sqrt{A_w}}{\sqrt{A_{w0}}} \quad (5.2)
\]

\[
N_{Layer} = \frac{L_0 * P_w}{\sqrt{A_{w0} * A_w}} \quad (5.3)
\]

5.1.2. Seraphim Estimation with blocked areas

Blocked areas such as thermal inserts or die pads under glued dies reduce the usable wiring area \( A_{w0} \). Therefore they enlarge the number of layers as calculated in equation (5.1). Here \( (A_{Block0}, A_{BlockN}) \) are the blocking area sizes and \( (L_{Block0}, L_{BlockN}) \) are the number of layers that are blocked from the corresponding area.

\[
N_{Layer} = \frac{L_0 \frac{\sqrt{A_{w0}}}{\sqrt{A_{w0}}} * P_w + \sum_{n=0}^{N} L_{Block_n} * A_{Block_n}}{A_w} \quad (5.4)
\]

5.2. New Routing Estimation

Often today’s published simple estimations such as Seraphim’s model [17], other estimations compared from Sandborn et al.[18] and Díaz-Alvarez [19] are not accurate enough. First, they are too optimistic because blocked areas from escaping are neglected. Second, they estimate the overall substrate size, but do not calculate the necessary width of a channel between components. In fact we experienced a misprediction of up to 20%.

Therefore we developed a new technique to estimate more accurately the routing limited size of a module: A routing analysis that calculates the resources available within the footprint area and predicts the necessary resources between the components. It is based on global routing. During global routing approximate paths for the wires are often determined on a channel graph as shown in figure 5.2(b). After this step the exact course for the wires along this channels, is determined in a detailed routing phase. The split into this two phases allows to derive the optimal paths without having to consider too many details and to avoid obstacles created by wires routed earlier during detailed routing. Thus, this strategy avoids the net ordering problem [67].

In this chapter the global routing objective is to minimize the substrate area for a given number of layers by maximizing the channel usage under the components and minimizing the spreading of the channels between the components. Our algorithm does not require a detailed netlist, but works with a simple netlist derived from a block diagram. It therefore works with as little information as possible. This enables a thorough analysis early in the design cycle when more information is not yet available.
5.3. GLOBAL ROUTING

Figure 5.1: Build-up Evaluation Method and Global Routing Steps (The steps are marked as rectangles and the input parameters are inside ovals)

5.2.1. Outline of the Algorithm

Our algorithm to generate and evaluate the suitable build-up for a system consists of the first three steps on the left side in figure 5.1 namely the footprint calculation, the chip limited size estimation and the routing limited size. They are the most important ones in a system analysis because the substrate primarily drives the cost and the size. The substrate area is either defined by the component footprints, the so called component limited size, or the interconnect requirements calculated because the area and number of layers required for escape and routing. Thus, to generate and analyze build-up alternatives we propose to follow the outlined strategy:

1. Calculate the component footprints for all first level interconnects and substrate technologies.
2. Derive the component limited size from the footprints.
3. Model the routing and set up a table for substrate size versus layer count.

In step 1 and 2 the minimum area required by the components and the minimum layers needed for escape is calculated. Then the necessary enlargement (if any) for different layer counts is calculated in step 3 using the new routing estimation procedure.

This new routing estimation is based on global routing. Its objective is the minimization of the substrate area for a given layer count. The estimation consists of several steps as shown on the right side in figure 5.1. It first generates a channel graph from a placement to model the wiring channels. Their capacities are then calculated to model the routing resources. Based on the channel graph possible routing paths, so called routing trees, are created for each net. Which trees to use is selected during global routing. The global routing can be formulated as integer program and estimates the required channel width between the components for a given number of layers. All global routing steps but the first one are repeated for different layer counts for all generic substrate technologies to draw a graph that shows the substrate size versus the number of layers.

5.3. Global Routing

Global routing is used to estimate the required routing resources and therefore the spacing between the components. After an introduction the cost function is defined in subsection 5.3.2. Then the
integer program can be formulated in subsection 5.3.3 and a solution method is described for the NP-hard problem in subsection 5.3.4.

5.3.1. Introduction

The global routing problem is to find approximate paths for each net in the design, so that the paths are non-intersecting and the area is minimized. The routing is performed on a channel intersection graph [68]. The edges in the graph correspond to the wiring channels, and the nodes correspond to the intersection of channels and electrical pins. Since in PCBs and MCMs the chip sizes and its pin locations are fixed or can be estimated, such a graph can always be constructed. The edges in the graph have a wiring capacity ($c(g)$) associated with them. It determines the maximum number of wires that can be routed through the corresponding wiring channel in the given technology. The wiring capacity is fixed for the channels below the components and for the ones through their footprints, further referred as constrained channels. The other channels, the further called global channels ($g$), are between the components. They have a relative capacity $c(g) = F(g) \cdot c_0(g)$. Here, $c_0(g)$ is the capacity of a global channel when the substrate size is the chip limited size. The capacity is calculated from its initial width $w_0(g)$. To ensure the routability of a design, the global channels are spread by factor $F(g)$ during global routing. $F(g)$ then allows to calculate the substrate enlargement factor $F$ as defined in subsection 5.3.2.

Once the routing graph is determined, the global routing problem can be formalized as follows [69]:

An instance of the global routing problem consists of a routing graph $G = (V, E)$, with vertices $V$ and edges $E$, and a set of supernets $N$, where each supernet is a subset of $V$. A supernet is a collection of identical nets (same terminal nodes e.g. whole or part of buses) that are treated as a single net with multiple wires. Each edge is labeled with a capacity $c : E \to \mathbb{R}^+$ and edge length $l : E \to \mathbb{R}^+$. Supernet $i$ has a multiplicity $d_i > 1$. For each of these nets $i \in N$, there is a set of admissible routes or trees $T_{i1}, \ldots, T_{iL}$. A solution to the global routing problem is a set of admissible trees, one or more for each net, so that the capacity $c(e \in E)$ on each edge is not exceeded by the traffic on that edge. The traffic $U(e)$ on an edge is defined by the weighted sum of all the routes that contain edge $e$:

$$U(e) = \sum_{i=1}^{N} \sum_{t=1}^{T_{iL}} y(i, t) \cdot a(e, t)$$

(5.5)

The weights $y(i, t)$ denote the number of wires in supernet $i$ that are routed using tree $t$, and $a(e, t)$ is a (0,1) function that specifies whether or not tree $t$ uses edge $e$.

The objective function minimized over all such feasible solution varies, depending on the design problem. Some formulations [70] try to minimize wire length, or the maximum ratio of the traffic on an edge to its capacity hoping to minimize the system area. In this thesis the objective is to minimize the substrate area for a given number of layers by maximizing the channel usage under the components (constrained channels) and minimizing the spreading $F(g)$ of the channels between the components (global channels). To achieve this the constraint global routing approach is used with a cost function that minimizes the substrate enlargement factor $F$. This factor grows linearly with the spreading of the global channels as described below. By minimizing the global channels the usage of constrained channels is up to 100% because the algorithm relieves the global channels by using the constrained ones.

Thus, our global routing problem is optimally solved by:
5.3. Global Routing

1. Finding a set of routing trees with short wire lengths and a high probability of meeting other constraints (see section 5.5)

2. and then minimizing the cost function while satisfying the edge capacity constraints (see subsections 5.3.2, 5.3.3, 5.3.4).

The cost function for our algorithm, the substrate enlargement factor \( F \), is defined below.

5.3.2. Substrate Enlargement Factor \( F \)

We define the substrate enlargement factor \( F \) as the multiplier of the chip limited substrate length and width to get the length and width of the final substrate size. Thus, the area grows with factor \( F^2 \). The minimum for \( F \) is one as the chip limited substrate size is the minimum \( F \) serves as cost function during global routing as its minimization results in a minimal area. Factor \( F \) is the maximum of the spreading factors \( F(\text{cut}) \) at any horizontal or vertical cut through the channel graph as shown in figure 5.2(b) and calculated in (5.8). \( F(\text{cut}) \) is the sum of all the spreadings \( F(g) \) of all global channels \( g \) in a cut. It is weighted by the ratio of their minimum channel width \( (w_0(g)) \) and the minimum substrate length \( \text{Length}_0 \) or substrate width \( \text{Width}_0 \) as shown in equation (5.7). We defined \( F(g) \) as the ratio between the traffic through an edge \( (U(g)) \) and its minimum channel width \( (w_0(g)) \) as shown in equation (5.6).

\[
F(g) = \frac{U(g)}{w_0(g)} \tag{5.6}
\]

\[
F(\text{cut}) = (F(g_1) - 1) + \frac{w_0(g_1)}{\text{Length}_0}
+ (F(g_2) - 1) + \frac{w_0(g_2)}{\text{Length}_0} + ... + 1 \tag{5.7}
\]

\[
F = \max(F(\text{cut}) \forall \text{cut}) \geq 1 \tag{5.8}
\]

For the independent (free) spreading of the channels equation (5.7) is formulated in (5.9). It is considered that global channels that are between the same components, i.e. in the same coarse channel as shown in figure 5.2(b), can not be spread independently. Therefore, their associated spreading factor \( F_{\text{max}}(g) \) is the maximum of their individual factors \( F(g) \).

\[
F_{\text{max}}(g) = \max(F(e) \forall e \in G \ni g)
\]

\[
F_{\text{free}}(\text{cut}) = \frac{\sum_{i=1}^{N_C} [(F_{\text{max}}(g_i) - 1) \times w_0(g_i)]}{\text{Length}_0} + 1 \tag{5.9}
\]

\( N_C \) is the number of channels in a cut. The spreading is flexible, but \( F_{\text{max}}(g) \) has to be calculated during integer program solving. This calculation can be avoided if all channels of a cut are spread with the same factor. This even spreading of the global channels is easier to calculate because the spreading factor \( F_{\text{even}}(\text{cut}) \) of a cut depends on the sum of all the minimum widths \( (w_0) \) of the channels in a cut \( (w_{0\text{tot}}) \). This sum can in advance be calculated per channel. To ensure that it is the same for all elements of a coarse channel \( G \), the minimum of all cuts through channel \( G \) is calculated in (5.10) previous to integer program solving. As \( F_{\text{max}}(g) \) is not calculated during global routing, equations (5.7) and (5.8) are reduced to (5.11):

\[
F_{\text{even}} = \max \left( (F(g) - 1) + \frac{w_{0\text{tot}}(g)}{\text{Length}_0} + 1 \right) \tag{5.11}
\]
In the following sections we will concentrate on even spreading for the calculation of $F$ to simplify the formulation. However, the used method can be extended to free spreading by the reader without much effort.

With the definition of the cost function in this subsection, we can below formulate the integer program.

5.3.3. Integer Programming Formulation

The global routing problem can be formulated as an integer program, by associating an integer variable $y_{ij}$ with tree $T^i_j$. Derived from equation (5.5), $y_{ij}$ is the number of wires in supernet $i$ routed with tree $j$. $a_{ij}^e$ is a (0,1) matrix that specifies whether or not tree $T^i_j$ uses edge $e$:

$$U(e) = \sum_{i=1}^{N} \sum_{j=1}^{j_{ij}} y_{ij} * a_{ij}^e$$

Then the global routing problem for even spreading (5.11) is given by the following integer program:

$$\begin{align*}
\text{Minimize} & \quad F = \max \left( \left( \frac{U(e)}{C_0(e)} - 1 \right) * \frac{\text{Width}(e)}{\text{Length}(e)} + 1 \right) \\
& \quad e = 1, ..., M, \\
\text{subject to} & \quad \sum_{j=1}^{j_{ij}} y_{ij} = d_i, i = 1, ..., N \\
& \quad U(e) < c_e, e = M_n + 1, ..., M \\
& \quad y_{ij} \geq 0
\end{align*}$$

$N$ is the total number of supernets, $d_i$ is the number of wires in net $i$, $j_{ij}$ is the number of routing trees for net $i$, $M_n$ is the number of the global routing channels (unconstrained) and $M$ is the total number of edges in the graph.

5.3.4. Solution Methods

Integer programming is, in general, NP-hard [71]. There are numerous ways of solving integer programs, e.g. cutting plane algorithms or branch and bound [69]. One method that has been shown to be very effective for solving the global routing problem is a randomized rounding applied to the linear relaxation of the integer program [72]. The basic idea is to relax the integer constraint in the formulation, which makes it a linear program and then to solve the linear programming problem. If the solution of the linear program is integral, we have an optimal global routing. If not, we need to transform it into an integer solution by rounding the non-integer values.

In our experience, solutions are often integer after solving the linear relaxation. In the cases with non-integer solutions, a simple rounding led to an enlargement of up to 2% for the spreading factor $F$. The result is therefore overoptimistic but we are on the safe side.

Before solving the linear program the algorithm needs a channel graph derived from the placement as described in the next section.

5.4. Channel Graph

The channel intersection graph needed for global routing is created from a given placement as described in 5.4.1. The assignment of capacities for the channels is based on the channel width (see subsection 5.4.2), the wiring pitch in the channels as described in subsection 5.4.3 and the footprints (section 4.1) and can be found in subsection 5.4.4.
5.4.1. Channel Graph

The coarse channels \((G \in E)\) between the components can be generated from a given placement (chip limited size) as shown in figure 5.2(a) and 5.2(b) between the big points \((P_G \in V)\). To obtain the needed fine channel graph, the channels \(G\) are split at the connections to the component connection points \((P \in V)\). The split points \((P_i \in V)\) are marked as small points. The component connection points \((P)\) on all four sides of a component are also referenced for the netlist. Thus, our algorithm does not need a detailed netlist, but works with a simple netlist derived from a block diagram. These points \(P\) have a twin point \((P_a \in V)\) inside the congested area. The channel \((P \leftrightarrow P_a)\) is needed to enforce the capacity limit through the footprint area. Here only passing connections are counted as the interconnections starting at these double points \((P, P_a)\) do not add to channel usage. In the center of a component the star point \((S \in V)\) merges the channels from all the twin points. For packages/first level interconnect, where the whole area under the component is blocked, the twin points and the star point become the same.

The above requirements can be fulfilled with a channel graph generated with the algorithm described below.

\[\text{Channel graph for Pentium MCM: The dies are white rectangles surrounded with the footprint area for wire bonding (gray). The spreadable global channels (}P_G \leftrightarrow P_G, P_G \leftrightarrow P_2, P_2 \leftrightarrow P_3, P_3 \leftrightarrow P_5\text{) are between the big points (}P_G\text{). They are split at the connection to a component footprint (}P_2\text{). The constrained channels are the connection from each component on all four sides to the global channels (}P_2 \leftrightarrow P\text{), the channels through the footprint (}P \leftrightarrow P_a\text{) and the channels (}P_a \leftrightarrow S\text{) under the components connected to the star point (}S\text{) in their center.}\]

5.4.1.1. Algorithm for Channel Graph Design

From any given placement the channel graph is created using the following, newly developed simple algorithm:

1. Draw lines between the components and on the surrounding rectangle until they hit a component or the rectangle border.
2. Merge nearby parallel lines or move both lines close the opposite sides if the spacing between is large.

3. These lines define the coarse channels $G$. The crossing points are the global points $P_G$.

4. Merge global points $P_G$ if no component is between four $P_G$ and if they are close.

5. Draw horizontal and vertical lines from the star point $S$ of each component until they hit a global channel. The crossing point with a global channel is a footprint connection point $P_g$ and the junction between the footprint border and these lines are netlist reference points $P$. These lines form the footprint connecting channels between $P_g$ and $P$, the through footprint channels between $P$ and $P_a$ and the to-star channels between $P_a$ and $S$.

6. Merge $P_g$ points that are close to each other and are between the same $P_G$ points.

7. Optional: If there is no component on the other side of the global channel then extend the line drawn in step 5 until it crosses a global channel from step 1 or connect the opposite $P_g$ if there is any.

A more sophisticated method for the channel creation is to replace steps 5f with:

5. The footprint connecting points $P_g$ are in the center between two global points $P_G$. Connect now the star point with all the $P_g$ around each component. At the junction of the connection and the footprint border are the new netlist points $P.x$ as shown in figure 5.3.

6. Split the nets assigned to the original $P$ according to their terminal's positions and assign them to the new $P.x$.

The advantage of the latter method is that the connection between two neighbors do not add to the usage of the orthogonal channel between them. However, this can create more than four netlist points per component and therefore requires exact location information of the component terminals. As the aim of this chapter is the ability to work with as few information as possible, the adoptions are left to the reader.

To calculate the channel capacities from the graph their widths have to be derived as described below.

5.4.2. Global Channel Width

The channel width for the constrained channels only depend on component data. Therefore there is no need to calculate their widths in advance to capacity calculation. For the global channels however the initial (minimum) channel width ($w_0$) needs to be derived from the placement. In addition to the spacing ($w$) between the footprints a part of the footprint can be used as calculated in (5.16) and as marked light gray in figure 5.2(a). In this area the layers needed for connecting the global channel with the footprint are subtracted in the capacity calculation. In (5.16) $O_{ESC}$ is the part of the escaping overhead $O_{ESC}$ of a footprint that can not be used for the global channel. For staggered vias the whole area outside the pad area can be used and $O_{ESC}$ is therefore often zero (5.14). But for through hole vias only the outermost via row can be saved as the others block the whole area underneath (5.15). The footprint without the global channel ($A_{ESC}$) is then calculated with (5.17).
5.4. Channel Graph

![Alternative Channel Graph](image)

Figure 5.3: Alternative Channel Graph

\[
O_{E_{sc2,separator}} = O_{E_{sc}} - 2 \cdot (L_{\text{space}} - D_{\text{viaLand}} - (k-1) \cdot d_{E_{sc}})
\]

(5.14)

\[
O_{E_{sc2,through}} = O_{E_{sc}} - 2 \cdot (\text{if}(k = 1: L_{\text{space}} + D_{\text{viaLand}}; d_{E_{sc}}))
\]

(5.15)

\[
w_0 = w + (O_{E_{sc,right}} - O_{E_{sc,left}})/2
\]

(5.16)

\[
A_{E_{sc2}} = (\sqrt{A_{\text{Path}}} + O_{E_{sc2}})^2
\]

(5.17)

For the width calculation of special global channels that are between other global channels, the space between them is equally distributed.

5.4.3. Channel Wiring Pitch

The channel wiring pitch depends on the substrate design rules such as line width \(L_{\text{width}}\), line space \(L_{\text{space}}\) and via land diameter \(D_{\text{viaLand}}\) as shown in figure 3.10. The line pitch i.e. \(L_{\text{width}} + L_{\text{space}}\) is often considered as routing pitch. But this is misleading because the vias lands are usually larger than the line width and therefore enlarging the routing pitch. A more appropriate calculation uses the number of lines \(L_{\text{Via}}\) that can be routed between two vias\((5.18)\).

\[
P_w = \frac{D_{\text{ViaLand}} + L_{\text{Via}} \cdot L_{\text{width}} + (L_{\text{Via}} + 1) \cdot L_{\text{space}}}{L_{\text{Via}} + 1}
\]

(5.18)

On today’s substrates the vias are not placed on a grid. So we propose to calculate the routing pitch as the average pitch of the via and the line pitch \((L_{\text{Via}} - 1)\). This calculation is based on the assumption that only one line is routed between two vias. For global channels the constraints can be relaxed to two lines between two vias \((L_{\text{Via}} = 2)\). It is a worst case model but it ensures a good routability and has been used with good results in our designs.
5.4.4. Channel Capacity

5.4.4.1. Global Channels

The capacity \( c(g) \) of the global channels \( g = (P_G \leftrightarrow P_G) | (P_g \leftrightarrow P_G) | (P_q \leftrightarrow P_q) \) is \( F(g) \ast c_0(g) \) as defined in subsection 5.3.1. Factor \( F(g) \) is estimated in the global routing step. \( c_0(g) \) is the original capacity and depends on the minimum channel width \( (w_0(g)) \), the number of layers \( (N_{Layer}) \) minus the number of layers required for escaping to this global channel \( (N_{Esc-Eff}) \) and the routing pitch \( (P_w) \):

\[
\begin{align*}
    c_0(g) &= (N_{Layer} - N_{Esc-Eff}) \ast \frac{w_0(g) - L_{Space}}{P_w} \\
    c(g) &= F(g) \ast c_0(g)
\end{align*}
\] (5.19)

5.4.4.2. Footprint Connecting Channels

The required capacity \( c_{raw} \) for the channels \( (P \leftrightarrow P_g) \), connecting the footprint with the global channels, is the capacity of the channel through the footprint \( c(P \leftrightarrow P) \) or \( c(P \leftrightarrow S) \) as described below plus the number of pins leaving from the footprint \( (5.20) \). The number of leaving pins is either taken from the component description or the netlist.

The necessary capacity \( c_{raw} \) can now be used to calculate the number of layers used to connect to the orthogonal global channel. Assuming the escaping length as a fourth of the global channel width \( (w) \), the average escaping width is the footprint without the global channel \( (\sqrt{A_{Esc2}}) \) widened by a fourth of both channel width right and left of the escaping channel. The number of escaping layers in this area is then shown in equation \( (5.21) \). With this number the real capacity of the channels \( (P \leftrightarrow P_g) \) is calculated in equation \( (5.22) \).

\[
\begin{align*}
    c_{raw}(P \leftrightarrow P_g) &= (C_N - C_P)/4 \text{ or real number} \\
    &+ [c(P \leftrightarrow P) \text{ or } c(P \leftrightarrow S)] \\
    N_{Esc-Eff} &= \sqrt{A_{Esc2} + \frac{w_{left}/4 + w_{right}/4 - L_{Space}}{P_w}} \tag{5.20} \\
    \text{if } N_{Esc-Eff} < N_{Layer}, \text{ then } N_{Esc-Eff} = N_{Layer} - 1 \\
    c(P \leftrightarrow P_g) &= \frac{N_{Esc-Eff} \ast \sqrt{A_{Esc2} + \frac{w_{left}/4 + w_{right}/4 - L_{Space}}{P_w}}}{P_w} \tag{5.21}
\end{align*}
\] (5.22)

5.4.4.3. Through Footprint Channels for Peripheral Interconnect or Area Array with empty center

The still undefined channel capacity through the footprint \( (P \leftrightarrow P_g) \) and under the footprint \( (P_h \leftrightarrow S), (P \leftrightarrow S) \) depends on the footprint type. For peripheral interconnects or area interconnects with an empty center, equation \( (5.23) \) can be used. Its capacity depends on the permeability of the footprint \( (Perm) \), the number of substrate wiring layers \( (N_{Layer}) \), the number of escaping layers of the footprint \( (N_{Esc}) \) and the width of the footprint without the global channel \( (\sqrt{A_{Esc2}}) \), the routing pitch including thermal vias if applicable \( (P_{w, Inside}) \) and the capacity of the channel under the component \( (c(P_h \leftrightarrow S)) \).

The capacity under a component depends on the width of the empty center \( (W_{open}) \) instead of the footprint width \( (5.24) \). Here the number of layers \( (N_{Layer}) \) is reduced by one if the attachment is wirebonding and the top layer under the die is a solid plane. As the escaping layers are only used to connect the terminals to routing channels leaving the component, these layers can be fully used from all sides. Therefore, the number of layers is \( \frac{N_{Layer} - 1}{2} - N_{Esc} + N_{Esc} \).
5.4. Channel Graph

If a component has no connections to one side the node \( P \) exists but there is no \( P_a \). In these cases the capacity of the connecting channel to the star point \( S \) is calculated as \( c(P_a \leftrightarrow S) \) in equation (5.24).

\[
c_{foot} = Perm \times (C_N/4 \text{ or real number})
\]

\[
c(P \leftrightarrow P_a) = \min [c_{foot} + (N_{Layer} - N_{Esc}) \times \frac{\sqrt{A_{Esc2}} - L_{Space}}{P_a \text{Inside}}]
\]

\[
c(P_a \leftrightarrow S) = c(P \leftrightarrow S) = \frac{N_{Layer} - 1 + N_{Esc}}{2} \times \frac{W_{open} - L_{Space}}{P_a \text{Inside}}
\]

5.4.4.4. Through Footprint Channels for Full Area Array

The capacity of full area array footprints (5.25) is very similar to the one with empty center (5.23). But as the points inside the footprint \( P_a \) merge with the star point \( S \) its interconnecting channel capacity has no more limits (5.26). The channel capacity within the footprint is then the permeability plus the capacity of half the remaining layers (5.25). If the resulting number of layers \( n \) is not in integral, it has to be rounded up for the direction with the heavier traffic and vice versa for the other direction.

\[
n = \frac{N_{Layer} - N_{Esc}}{2}
\]

\[
c(P \leftrightarrow P_a) = c_{foot} + n \times \frac{\sqrt{A_{Esc2}} - L_{Space}}{P_a \text{Inside}}
\]

\[
c(P_a \leftrightarrow S) = \infty
\]

5.4.4.5. Permeability of a Footprint

The permeability of a footprint serves in the routing analysis as measure for the routing congestion of the area below and beside a component.

Often the escaping from a component does not block the footprint area completely. The capacity of the routing channels through this congested area depends on the number of traces that can pass between the pads additionally to the escape routing. Based on the assumption that the blocking of a footprint is evenly spread, we define the permeability of a footprint \( Perm \) as the average number of lines that can pass between two pads. This number depends on the pad pitch \( P_{Pitch} \), the pad width \( P_{Width} \), the design rules such as line width \( L_{Width} \) and line space \( L_{Space} \) and the number of layers used for the escaping \( N_{Esc} \) of a component as shown in equation (5.27).

For peripheral lead packages such as QFPs and TAB, \( Perm \) must be integer. Wire bonding however, has the advantage that the pads can be moved to allow the full utilization of the spacing between. Therefore, there is no need to floor the calculated permeability as it was required for packages with peripheral leads. The pad pitch \( P_{Pitch} \) for wire bonding can be estimated as the average pad pitch on the IC \( \bar{P}_{PP} \) multiplied by the number of wire bond rows \( N_{Row} \). For components with area pin configuration the permeability is zero as their layer usage is maximized in order to minimize the number of layers for escaping.

\[
Perm_{Peripheral} \leq N_{Esc} \times \frac{P_{Pitch} - P_{Width} - L_{Space}}{L_{Width} + L_{Space}}
\]
Now the channel graph is defined and the netlist can be mapped onto it. First several feasible trees are generated for each net as described in the section below. Then the selection which of these trees to use is performed in the global routing step.

5.5. Tree Generation

The first step in solving global routing problems is to find a set of routing trees for each net that satisfies electrical constraints such as maximum length, topology or other. There has been considerable research on tree generation algorithms in the past, primarily focused on the Steiner tree generation problem, with the objective of minimizing total wire length [69]. However, even for simple wire length minimization objective, the optimal Steiner tree construction problem is NP-hard [71]. Fortunately, the nature of the problem allows us to do quite well with heuristic solutions. Firstly, compared to on-chip nets, nets on a PCB or MCM have a smaller fanout. Secondly, the required routing resources do not have to be absolutely minimized as routing under the components is possible and the minimization of the total wiring length does not necessarily lead to the smallest possible size. Hence the freedom allowed by relaxing the wire length minimization objective makes the tree generation tractable. The key to constructing feasible routing trees is to generate for each net a set of alternative trees with small wire length that have no common edge and to let the global routing chose some of them. Here Manhattan style length calculation was used even when the effective channel length can be shorter for channels connecting orthogonally to a common channel as shown in figure 5.4.

For point-to-point nets, there is only one topological way of constructing routing trees. Here, the shortest paths from the driver to the receiver and little longer alternatives need to be found as described below in section (5.5.1). For multi-point nets several topologies such as steiner tree, daisy chain, far-end and near-end cluster can be used. As the purpose of this chapter is routing area minimization, minimum steiner trees are used in the remainder of the paper. Other topologies might be used to fulfill other requirements. The details and the creation of alternative trees is described in section 5.5.2.

![Figure 5.4: Comparison of calculated length (solid line) and effective length (dashed line)](image_url)

5.5.1. Point-to-point nets

For point to point nets, the routing trees are constructed by following the shortest path from the driver pin to the receiver pin.

**Def.** The *shortest* path between two vertices 'u' and 'v', in a weighted graph G is defined as the path \( u \rightarrow v_1 \rightarrow v_2 \ldots \rightarrow v \) so that \( \sum_i w(v_i, v_{i+1}) \) is minimal.
There is extensive literature on solving the shortest path problem. See [73], [74] and chapter 3 in [69] for a good review of existing literature. Lengauer [75] and Kote [76] describe a generalized extension of the shortest path problems.

However, as the aim of this thesis is the area estimation the tree generation can be relaxed to the problem of finding short paths. The tree generation is solved by finding several paths with short wire length. This procedure helps keeping the problem quite tractable.

The problem to be solved is to enumerate the k shortest paths for a given pair of source and destination vertices. A restriction to be imposed on these paths is that there should be no repeated vertices along any of the paths because a path length can be extended by adding a small cycle in the path. The cycle acts only as a shorted loop in the interconnect and does not reduce the congestion. An algorithm to solve the k shortest path problem, with no repeated nodes, is given in Lawler [73]. This algorithm is described next with an illustration.

The required data structure is a list of shortest paths called $P$. The source vertex is called $v_1$ and the destination is $v_n$. Initially, $P$ is empty. The shortest path computation proceeds as follows:

1. Compute the shortest path from $v_1$ to $v_n$ using Dijkstra’s shortest path algorithm [77]. Place this path on $P$ and set $m = 1$.

2. If $P$ is empty then stop. There are no more paths between $v_1$ and $v_n$. $V$ collected all the shortest paths in ascending order. Otherwise, remove the shortest path from $P$ and put it into $V$ as the $m$th shortest path, $V_m$.

3. Computing $k$ paths asks for forcing the short path procedure to avoid using certain edges in the graph. Suppose that $V_m$ contains the vertices $v_1, \ldots, v_{n-1}, v_n$, and that $V_m$ is the shortest path from $v_1$ to $v_n$ subject to the condition that it is forced to go through vertices $v_1, \ldots, v_p$ where $p \leq n - 1$ and that certain edges from $v_p$ were excluded in doing this shortest path computation (This information is stored with $V_m$ as it was done for the same entry in $P$).

   If $p = n - 1$, find the shortest path from $v_1$ to $v_n$ subject to the condition that edges $(v_1, v_2), (v_2, v_3), \ldots, (v_{n-2}, v_{n-1})$ are included and that $(v_{n-1}, v_n)$ is included, in addition to the other edges excluded in calculating $V_m$. If such a path exists, then place it in $P$ along with a record of the conditions under which it was obtained.

   If $p < n - 1$, then find the shortest path from $v_1$ to $v_n$ subject to the following sets of conditions:

   (a) Edges $(v_1, v_2), (v_2, v_3), \ldots, (v_{p-1}, v_p)$ are included and edge $(v_p, v_{p+1})$ is excluded, in addition to the edges excluded in calculating $V_m$.

   (b) Edges $(v_1, v_2), (v_2, v_3), \ldots, (v_{p+1}, v_{p+2})$ is excluded, in addition to the edges excluded in calculating $V_m$.

   ;

   (x) Edges $(v_1, v_2), (v_2, v_3), \ldots, (v_{n-2}, v_{n-1})$ are included and edge $(v_{n-1}, v_n)$ is excluded, in addition to the edges excluded in calculating $V_m$.

   If such paths exist, then place them in $P$ along with a record of the conditions under which they were obtained. Increment $m$ and return to step 2.
Figure 5.5 shows a graph on which the 3 shortest paths from vertex A to vertex F have to be computed. Table 5.1 shows the paths in list $P'$ at the beginning of step 2 in the algorithm for several iterations.

5.5.2. Multi-point nets

For multi-point nets several topologies can be used as illustrated in figure 5.6. If the selection is not restricted by electrical constraints Steiner trees are used. They minimize the tree size and therefore need less routing resources than other topologies. For the creation of several alternatives for the found steiner tree, the algorithm for the point-to-point[73] nets was adapted as follows:

1. Compute the shortest spanning tree for vertices $v_1$ to $v_n$ using Kruskal's algorithm to build a Steiner tree [69] chapter 3.10.3. Place this tree on $P$ and set the pass number $s = 0$.

2. If $P$ is empty then take all trees from $V$ that have $s$ or more deleted channels and put them into $P$. Then increment pass number ($s$).

3. If $P$ is empty, then stop. There are no more spanning trees for $v_1$ to $v_n$. $V$ holds the valid configurations. Otherwise, remove the shortest tree from $P$ and put it into $V$ as the last tree, $V_l$.

4. Computing $k$ trees asks for forcing the Steiner tree procedure to avoid using certain edges in the graph. Suppose now, different to point-to-point nets, that $V_l$ contains the edges $e_1, ..., e_l$, spanning the minimum tree for the vertices $v_1$ to $v_n$, subject to the condition that certain edges where deleted from the routing graph. These deleted edges are stored in $V_l$ as it was done for the same element in $P$. It is also stored up to which edge number
5.6. Case Study

A typical industrial PC based on a "Pentium" CPU [44] as presented in chapter 3 is used to demonstrate the new routing estimation. Wire bonding was chosen as first level interconnect because of the tight pitch of the Pentium and TSC die and its ability to spread the substrate connections onto several rows. For the substrate selection first the chip limited size was estimated for all types of substrates summarized in table 5.2 as described in the next subsection 5.6.1. Then a routing analysis was performed using Seraphims estimation and the novel global routing approach in subsection 5.6.2 and 5.6.3.

5.6.1. Chip Limited Size

The chip limited size is 32*32 mm for the thinfilm solution. It is enlarged to 40*37 mm for a ceramic and 40*40 mm for an SBU substrate as shown in figure 3.7 and table 5.4. PCB was not considered because it is impossible to wirebond the dies onto it, due to the several pad rows that would be required on the substrate and their large spacing caused by the vias required to escape from the inner bondpad rows. The MCM-D substrate only needs two layers for escaping from the
ICs, the others (MCM-C and SBU) require three.

### 5.6.2. Seraphim’s Estimation

For Seraphim’s estimation [17] the minimum size, the chip pitch ($P_D$) and the number of I/O terminals are have to be known. For the Pentium MCM, these parameters and the results are listed in table 5.3. Here, the connections from the substrate to the package are not included in order to simplify the chip pitch calculation. $P_D$ is different from the value in chapter 3 because it is calculated from the sum of area and spread as explained in subsection 5.1.1.

It can be seen that the MCM-D solution seems easily routable on the chip limited size, having a good safety margin when using two layers for the implementation. The same applies for the ceramic substrate with six layers, but the SBU alternative might need six layers instead of the estimated five. This uncertainty is due to the fact that Seraphim’s estimation neglects local congestions and is therefore too optimistic. The layer reduction forces a size enlargement calculated here without any safety margin.

### 5.6.3. Global Routing

For the global routing estimation a channel graph was generated for the Pentium MCM as shown in figure 5.2(b). The netlist was taken from the design and mapped to the connection points ($P'$) of the ICs and the package. Using this input data the substrate enlargement was calculated for several layer counts as shown in table 5.4.

Whereas the module is routable on the chip limited size for MCM-D and C substrates, the SBU solution requires slight spreading. To obtain a layer reduction the substrates have to be further enlarged.

### Table 5.2: Substrate Parameters

<table>
<thead>
<tr>
<th>Substrate Type</th>
<th>$L_{\text{width}}$ [$\mu$m]</th>
<th>$L_{\text{space}}$ [$\mu$m]</th>
<th>$D_{\text{ViaLand}}$ [$\mu$m]</th>
<th>$P_{w-\text{eff}}$ [$\mu$m]</th>
<th>$B P_{\text{width}}$ [$\mu$m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCB</td>
<td>125</td>
<td>125</td>
<td>650</td>
<td>510</td>
<td>125</td>
</tr>
<tr>
<td>SBU</td>
<td>15</td>
<td>15</td>
<td>300</td>
<td>260</td>
<td>125</td>
</tr>
<tr>
<td>MCM-C</td>
<td>125</td>
<td>125</td>
<td>200</td>
<td>290</td>
<td>125</td>
</tr>
<tr>
<td>MCM-D</td>
<td>20</td>
<td>30</td>
<td>50</td>
<td>65</td>
<td>100</td>
</tr>
</tbody>
</table>

### Table 5.3: Seraphim’s Estimation Pentium MCM

<table>
<thead>
<tr>
<th>Substrate</th>
<th>$P_D$ [mm]</th>
<th>$N_T$</th>
<th>$N_{\text{Layer}}$</th>
<th>$\text{Size}$ [mm$^2$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBU</td>
<td>14</td>
<td>826</td>
<td>4.9</td>
<td>40*40</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>49*49</td>
</tr>
<tr>
<td>MCM-C</td>
<td>13</td>
<td>826</td>
<td>5.5</td>
<td>40*37</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>43*40</td>
</tr>
<tr>
<td>MCM-D</td>
<td>10</td>
<td>826</td>
<td>1.5</td>
<td>32*32</td>
</tr>
</tbody>
</table>
5.6. Case Study

Table 5.4: Results of Global Routing Estimation for Pentium MCM

<table>
<thead>
<tr>
<th>Substrate</th>
<th>Chip Limited [mm(^2) mm]</th>
<th>(N_{layer})</th>
<th>(F)</th>
<th>Estimated Size [mm(^2) mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBU</td>
<td>40*40</td>
<td>5</td>
<td>1.04</td>
<td>41.6*41.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>1.26</td>
<td>50.4*50.4</td>
</tr>
<tr>
<td>MCM-C</td>
<td>40*37</td>
<td>6</td>
<td>1.00</td>
<td>40*37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>1.1</td>
<td>44*41</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>1.3</td>
<td>52*48</td>
</tr>
<tr>
<td>MCM-D</td>
<td>32*32</td>
<td>2</td>
<td>1.01</td>
<td>32.3*32.3</td>
</tr>
</tbody>
</table>

Table 5.5: Comparison of Size Estimation Methods. The estimated routing area is compared relatively to the chip limited size. If the routing area is smaller than or 100% the substrate size is the chip limited size.

<table>
<thead>
<tr>
<th>Substrate</th>
<th>Final Size [mm(^2)]</th>
<th>(N_{layer}) Signal</th>
<th>Routing Area Seraphim</th>
<th>Global Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBU</td>
<td>n/a</td>
<td>5</td>
<td>98%</td>
<td>108%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>132%</td>
<td>159%</td>
</tr>
<tr>
<td>MCM-C</td>
<td>n/a</td>
<td>6</td>
<td>93%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>116%</td>
<td>122%</td>
</tr>
<tr>
<td>MCM-D</td>
<td>32*32</td>
<td>2</td>
<td>75%</td>
<td>101%</td>
</tr>
</tbody>
</table>

5.6.4. Comparison of the Routing Estimation Methods

Seraphim’s estimation provides an overall routing estimation neglecting local congestions. It is therefore a lower bound for the substrate size as shown in table 5.5. In fact Seraphim’s estimations are 6% to 27% smaller than our algorithm. The global routing estimation is much more accurate as it considers the congestions. Furthermore, it uses a netlist to know the necessary routing instead of estimating it. The global routing estimation therefore does not require a safety margin as is necessary for Seraphim’s estimation. As manual optimizations can relieve a congested channel, the global routing estimation is an upper bound for the substrate size. The presented example and other studies showed that Seraphim’s estimation requires a safety margin up to 20% and that the global routing approach provides an accuracy of better than ±5%.
JavaCAD

The strategies presented in the previous chapters allow a highly structured approach by virtually building and analyzing several implementations. However, it is a time consuming task because a lot of manual analysis and interactions have to be performed. Thus, an implementation of an MCM design advisor is required.

Input for the advisor is a machine readable form of the system description including the system specification, for example VHDL, or the components are already fixed. When the components are fixed, they are grouped into functional partitions and the packaging evaluation can be started in a packaging advisor as shown in figure 6.11. We concentrated our activities on the packaging advisor because there are a lot of activities to transform a VHDL description into one or several ICs [11], [12], [78]. The packaging advisor creates several build-up alternatives by performing a design space exploration as shown in figure 3.3. After considering all the technical aspects the final decision can be taken after cost modelling in a cost-performance trade-off. To support this process, Thomas Bruhin [58] implemented a java based tool, the JavaCAD, in his masters thesis.

JavaCAD implements the dark gray parts of the packaging advisor in figure 6.11. The tool consists of a frontend where Thomas implemented the footprint models and integrated the global routing. It also includes a simple cost estimation as well as an interface to the cost modelling tool MOE [57].

6.1. JavaCAD

JavaCAD is based on the methodologies and models described in chapters 3 to 5. It implements all the equations and has been used to check them. JavaCAD allows to describe a project with very few parameters. Providing input of all the components in the project window, the remaining data such as assembly data and substrate design rules are taken from libraries and a placement is calculated. If a component is already in the library its data is taken from there. The placement can then be edited and Seraphim’s estimation provides a first hint of the number of layers required for interconnecting the components. When the user provides a simplified netlist, the tool performs the global routing described in chapter 5 and provides more accurate data for substrate size and number of routing layers needed. To support this process, the channel graph is generated automatically from the edited placement.

JavaCAD provides a graphical user interface as described in the next section.
6.2. User Interface

JavaCAD interacts with the user through five windows, namely the project window, the placement window, the layer graph, the component data summary and the component footprint comparison as shown in figure 6.1. The main window is the project window where a component set (partition) is managed. In the placement window, the position and the angle of the components can be edited or placed automatically. Additional data is taken from the libraries described in section 6.6. The other windows are result windows. All windows are described in the following subsections.

Figure 6.1: JavaCAD User Interface

6.2.1. Project Window

In the project window (figure 6.2) one can manage the different components of a partition as well as change the substrate technology. One row represents one component. On the left side the input data such as number, sizes, bonding method and pad counts can be edited as detailed below. On the right side the calculated data such as size and layer count is fed back as detailed below. Optional data such as cost, yield and ID for
the MOE interface is edited on the very right side.
The buttons have the following functions:

- “MCM-D” is the substrate selection combo box selecting actual MCM-D substrate technology
- “Add Component”, “Copy Component”, “Delete Component” to manage the component list
- “Edit Comp Data” is used to change the assembly method of the selected component and to specify assembly specific component data.
- “Show Comp Data” summarizes the most important calculated data of the selected component.
- “Component sizes” displays a footprint size comparison for the selected component, mounted with all assembly methods on all substrates specified in the substrate library.
- “Calculate” forces recalculation of the result data. Normally the results are automatically updated after an input change.
- “Layer” displays a graph where the number of layers vs. substrate edge length is drawn.
- “Placement” opens the placement window.

![Figure 6.2: Project window](image)

- **Table:** Component sizes.
In the project window a component is characterized by:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>Component name</td>
<td>String</td>
</tr>
<tr>
<td>RefDes</td>
<td>Component number</td>
<td>String</td>
</tr>
<tr>
<td>Length</td>
<td>Chip or package length</td>
<td>( \mu m )</td>
</tr>
<tr>
<td>Width</td>
<td>Chip or package width</td>
<td>( \mu m )</td>
</tr>
<tr>
<td>Height</td>
<td>Chip or package height</td>
<td>( \mu m )</td>
</tr>
<tr>
<td>Padlocation</td>
<td>Position of the pads</td>
<td>all, short, long, area (see figure 6.3)</td>
</tr>
<tr>
<td>Assembly</td>
<td>First level interconnect or package</td>
<td>Wirebond, TAB, FC or package name</td>
</tr>
<tr>
<td>Cpp</td>
<td>Pad pitch</td>
<td>( \mu m )</td>
</tr>
<tr>
<td>Padwidth</td>
<td>Pad width</td>
<td>( \mu m )</td>
</tr>
<tr>
<td>Cn</td>
<td>Number of chip or package pads</td>
<td>1</td>
</tr>
<tr>
<td>Cp</td>
<td>Number of power pads</td>
<td>1</td>
</tr>
<tr>
<td>Cost</td>
<td>Component cost</td>
<td>cents</td>
</tr>
<tr>
<td>Yield</td>
<td>Component yield</td>
<td>promille</td>
</tr>
<tr>
<td>ID</td>
<td>ID for the MOE interface</td>
<td></td>
</tr>
</tbody>
</table>

The length is not necessarily the longer side of a component. To select the position of pin number 1, the following definition is applied: Pin 1 is in the left lower corner if the width is horizontal and the length vertical as shown in figure 6.4.

From the above input and the selected substrate the following parameters are calculated:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFootprint</td>
<td>Footprint size</td>
<td>( mm^2 )</td>
</tr>
<tr>
<td>AF/A</td>
<td>Ratio footprint to component size</td>
<td>1</td>
</tr>
<tr>
<td>Perm</td>
<td>Permeability of the footprint</td>
<td>1</td>
</tr>
<tr>
<td>Sig Layers</td>
<td>Number of signal layers required for escaping</td>
<td>1</td>
</tr>
</tbody>
</table>

![Figure 6.3: Padlocation on an IC](image)
6.2. User Interface

6.2.2. Placement

The placement window in figure 6.5 shows the component placement. The grey rectangle in the middle represents the die, whereas the white area is needed to connect the die to the substrate (= overhead).

The "Sum of Area" is the sum of all component areas (Sum of all Aesc in the project-table.) The "Sum of Area" is smaller than a placement such as a generated Simple Placement, because it does not include area lost through non-ideality of component shapes. This underestimation is reduced by adding 10% to the calculated area (rule of thumb).

The placement can be edited to include the system designer's knowledge. The components can be moved except the largest one, which acts as reference. They might also rotated. The substrate size and cost are updated online. This placement is the basis for the layer graph, where the number of layers vs. substrate size is shown based on Seraphims estimation. Furthermore, the routing graph is shown in the placement window for the global routing estimation. It is exported with the "Node" button.

The above functions are controlled with the following buttons:

- "Autoplac" generates a Simple Placement as described in subsection 4.3.2.
- "Nodes" exports the routing graph for global routing.
- "Rotate" rotates the selected component by 90 degree.
- "Zoom in/out" zooms the placement.

From this placement the layer graph is calculated and displayed in the corresponding window as described below.

6.2.3. Layer Graph

The layer graph shows how many layers (y-axis) are needed and how much the substrate has to be enlarged (x-axis) to save one layer. The x-axis of the graph starts at the smallest possible substrate edge length defined by sum of areas of the component footprint. The upper curve shows the estimation result from the edited placement (with Position) and the lower one is estimated from sum of areas. Both curves are only valid for substrate edge lengths larger or same than their minimum.
The calculation parameters for Seraphim’s estimation such as chip pitch \( P_D \), wiring length \( L \), wiring pitch \( P_{\text{wire}} \) and number of I/Os \( N_T \) are displayed above the graph derived from the “the sum of areas”, and below the graph calculated from the edited placement.

![Graph](image)

**Figure 6.5: Placement Window**

If a netlist is generated from a block diagram or simplified from a schematic entry, a global routing can be performed to get more accurate results as described below.

### 6.3. Global Routing

At present the global routing is performed in a separate Java program. It requires the following inputs, which all but the netlist are exported from the JavaCAD:

- Component placement and description (Export in File menu)
- Routing channel graph (Export in Placement Window as shown in figure 6.7)
- Netlist
6.4. COST MODELLING

By means of this input files further described below, the global routing tool creates an input file for a linear programming solver, runs it and interprets its output. The global routing results are then fed back into the JavaCAD.

![Channel Graph for Global Routing](image)

Figure 6.7: Channel Graph for Global Routing

6.3.1. Node Numbering

The channel graph consists of the node types shown in figure 6.8(a). In the center under the component the star point \( S \) is situated. Around it on four sides the \( A \) points can be found on the die edges. They are the twin points of the \( P \) points situated on the border \( \Omega_{ext2} \) (see section 5.4.2) and are referenced in the netlist. The described three node types define a component and are numbered as shown in figure 6.8(b). The component points are numbered from 1 to 999. The star point is the first number of a component starting at the first component in the project window. The \( P \) and \( A \) points are then numbered clockwise starting on the side where pin 1 is situated. They both have the same number because they are the same points for the netlist. To distinguish them in the graph, the \( A \) points have the suffix \( a \).

The \( g \) points are the junction of the channel from a component and a global channel (fat line). At crossings of the global channels the \( G \) points can be found. They are numbered from 1000 to 1999 starting at the left lower corner of the channel graph and are not used in the netlist. Furthermore, if the \( g \) or \( G \) points are situated at the substrate edge, they are connected to a \( P \) point outside the substrate to model the connections to the package. These \( P \) points are numbered counterclockwise from 2000 to 2999 starting at the lower left corner as shown in figure 6.8(b). If there are connections to the outside they are referenced in the netlist.

Now the parameters of a module are calculated. To select the final implementation a cost modelling is mandatory as described below.

6.4. Cost Modelling

For the cost estimation each component, substrate and assembly method requires associated cost and yield. From these cost parameters and the calculated sizes, layer counts and assembly cost, an online calculation is done.

For a detailed calculation including modelling of the assembly process, the input data and the calculated data are exported to the ETH cost modelling tool MOE[57]. The MOE tool simulates
Chapter 6: JavaCAD

the assembly of a module and outputs the cost of a working module which is fed back into JavaCAD. Furthermore, MOE allows to analyze reasons for yield loss or to optimize the assembly process.

After these calculations the JavaCAD exports the results into ASCII files as described in the next section.

6.5. Data Export

JavaCAD exports the results of its calculations for use in CAD systems:

- Placement coordinates including angle
- Keepout areas around components
- Component pad coordinates and calculated wire bond pads on substrate or CSP pad coordinates
- Channel Graph, selected routing tree and spread placement.

This data is exported into ASCII csv files. They can be viewed in MS Excel and be interpreted by a layout tool. Thus, the raw data can be used by every layout tool if a macro or plugin is available. Figure 6.9 shows a layout of a wirebonded IC where the wirebond pads were calculated in JavaCAD and then imported into the layout tool Protel.

6.6. Libraries

In addition to component parameters JavaCAD requires substrate, assembly and package data for the footprint calculation and the routing estimations. Depending on the selected first level interconnect either die data or package data are used for the calculation.

To address the above requirements four libraries (Die/Component, Package, Substrate and Assembly) were set up as shown in figure 6.10. These libraries are maintained in an ACCESS database and exported in ASCII csv-format for the JavaCAD.

In table 6.1 the implementation of the required libraries is summarized. It can be seen that the die and the component tables are merged into one file. This ensures the consistency of the DIE
Table 6.1: JavaCAD Libraries

<table>
<thead>
<tr>
<th>Access Table</th>
<th>Access Query</th>
<th>Export</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIE Component</td>
<td>Comp.DIE</td>
<td>componentus.csv</td>
</tr>
<tr>
<td>Package</td>
<td></td>
<td>package.csv</td>
</tr>
<tr>
<td>Substrate</td>
<td></td>
<td>substrate.csv</td>
</tr>
<tr>
<td>Assembly</td>
<td></td>
<td>assembly.csv</td>
</tr>
<tr>
<td>Pins</td>
<td></td>
<td>Comp.package_pins.csv (Optional)</td>
</tr>
</tbody>
</table>
data per component. Furthermore, the Access table "pins" is optional for calculations based on components' pad coordinates.
The libraries are further detailed in the following subsections. Some parameters are marked as optional and the parameter's name in ACCESS is written in parenthesis.

6.6.1. DIE Library

ACCESS table: DIE
In this table all relevant data for a die is stored

- Name and an optional description (Name, Description)
- DIE-Sizes (Length, Width, Height)
- Minimal pad pitch (Padpitch)
- Number of pads (Cn), number of power pins (Cp) and not connected pins (Cnc)
- Pad location (Padlocation) valid entries are: short, long, area or peripheral as shown in figures 6.3(a) to 6.3(c).
- Optional pad sizes (Padlength, Padwidth)
- Optional distance pad center to chip edge (P2C)
- Optional distance pad center to chip edge on both sides (P2S)
- Array data (Nl, Ns, Nle, Nse) only used for area array (Padlocation=Area). 
  \( N_l \) = Pad count on long chip edge, \( N_s \) = Pad count on smaller chip edge.
  For an empty center the following parameters are required:
  \( N_{le} \) = Missing pad count on longer chip edge.
  \( N_{se} \) = Missing pad count on smaller chip edge.
- Optional maximum junction temperature (TJ)

6.6.2. Component Library

ACCESS table: Component Library
This table combines the die data with package and other data
6.6. Libraries

- **ID**
  Unique ID of a component (used as reference in the database)

- **pDIE, pPackage**
  pDIE is the component name used in the DIE table and pPackage is the package name. p at the beginning of the name stands for a reference to another table. Thus, pPackage is a reference to the field Name of the Package table. A bare die entry is marked as pPackage = DIE.

- **Power Pins (Pp) and Not connected Pins (Pnc) of the package.**
  Only required if not a bare die entry.

- **Optional (Cost) and (Yield)**

- **Optional (Date) for the last update**

- **Optional (obsolete) yes/no field**

6.6.3. Package Library

ACCESS table: Package

The package library describes the packages of the packaged components in the component library. Furthermore, these packages can also be used to package an MCM substrate.

**Required:**

- **Name and optional description** (Name, Description)

- **Package sizes** (Length, Width, Height)

- **Minimal pad pitch** (Padpitch), Inner lead pitch (InnerPitch)

- **Number of pins** (Pn)

- **Pad location**

- **Pad sizes** (Padlength, Padwidth)

- **Array Data for area array packages** (Nl, Ns, Nle, Nse, Nli, Nsi)
  Additional to the parameters described in the DIE table an array in the empty center is characterized as:
  
  \[ N_{lf} = \text{pad count on the longer side in the center} \]
  
  \[ N_{sf} = \text{pad count on the smaller side in the center} \]

**Optional:**

- **Cavity size** (Cavlength, Cavwidth)

- **Thermal junction to case resistance** (RJC)

- **Dielectric permittivity** \( \varepsilon_r \) of encapsulation (erMold)
6.6.4. Substrate Library

ACCESS table: Substrate
In this table the available substrate technologies are parameterized.
Required:

- Name and optional description (Name, Description)
- Line width and space (Lwidth, Llength)
- Via land diameter (Dvia_land)
- minimal bond pad width (BPwidth)
- Via type (through): Yes if through hole via
- Maximum signal layers and total layers (SignalLayers, TotalLayers)
- Vias in pads (Pad_vias): Yes if allowed
- Number of lines between vias for escaping (Lescape) for through hole vias only
- cost per area \([cm^2]\) and layer (cost)
- Distance from footprint to substrate edge (C2E)

Optional (Not used yet):

- Substrate thickness (thickness)
- Metal thickness (t)
- Dielectric height (h)
- Dielectric constant (er)
- Metal type (metal)

6.6.5. Assembly Library

ACCESS table: Assembly

In this table only four entries namely FC, SMD, TAB and wirebond are allowed. BGAs and other area array packages are calculated with the SMD parameters. Their parameters can be adapted but no additional entries can be added as JavaCAD actually uses only these four assembly methods. Their parameters are:

- Assembly type (Type) FC, SMD, TAB or Wirebond
- Distance chip to chip (C2C)
  This parameter is given from the placement machine or die bonder, which require a minimal space between any components.
- Pad sizes (Padwidth, Padlength)
- Bondpad to bondpad for adjacent ICs (Bspace)
• Inner and Outer lead pitch (ILP, OLP)
• Vias in pads allowed (Padvias): Yes if allowed
• Minimum distance from IC to metal for gluing (C2P)
• Maximal and minimal wire bond length (1max, 1min)
• Loop height of bond wire above IC (HLoop), typical 150 µm
• Maximum angle between wire and on-chip bondpad (BCP), typical 45 degree
• Maximum angle between wire and substrate bondpad (deltaBetaBP), typical 10 degree
• Optional (Cost) and (Yield) per component for SMD and FC and per pin for TAB and wirebond

Obviously not all the parameters are used for all assembly methods. Tables 6.2 and 6.3 describe the requirements.

### Table 6.2: Assembly Parameters vs. Assembly Methods

<table>
<thead>
<tr>
<th>Assembly</th>
<th>C2C</th>
<th>Padwidth</th>
<th>BPlength</th>
<th>ILP.OLP</th>
<th>Padvias</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMD</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FC</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>TAB</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Wirebond</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

### Table 6.3: Wirebond only Assembly Parameters

<table>
<thead>
<tr>
<th>Bspace</th>
<th>C2P</th>
<th>Lmax.Lmin</th>
<th>HLoop</th>
<th>BCP</th>
<th>dBBP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

### 6.6.6. Pins Library

ACCESS table: Pins

This table optionally holds the pad coordinates of a component.

• unique component ID (pID) from component table
• Pad number (number)
• Signal name (signalname)
• x coordinates based on (0,0) in chip center (x)
• y coordinates (y)
• Pin location (location) “top, right, bottom, left”
• optional: Pin width (width)

The pins of a component have to be sorted counter clockwise.
6.7. Summary and Outlook

JavaCAD implements the most important parts of a HDP packaging selection process. It starts with minimum input data and combines the product specification with library data to generate and evaluate implementation options. However, the implementation of the process is not complete as shown in the design flow in figure 6.11. This design flow is proposal to automate the design space exploration described in chapter 3 and illustrated in figure 3.3.

Up to now, JavaCAD provides the steps footprint generation, component limited size, routing limited size and cost modelling marked as dark gray rectangles in figure 6.11. The tool could be enhanced to interact with commercial tools allowing a structured data interchange. Layouted parts and data gained from simulations should be fed back.

For the proposed enhancements, marked as white rounded rectangles in figure 6.11, some ideas can be found in literature. John [79] proposes a method to automate signal analysis as well as EMC simulation. Descriptions of trade-off analysis can be found in [80] and [81] and some effort to automate layout were done at IMEC [82].

Nevertheless, JavaCAD is ready for utilization and was used to perform the case studies described in the next chapter.
6.7. Summary and Outlook

Figure 6.11: Proposed automated MCM design flow
In this chapter two case studies are used to demonstrate the benefits of the algorithms presented in chapters 4 and 5 and to prove them. In the first section an MP3 Player is designed. The models were mainly used to select the suitable substrate technology. The world’s smallest GPS receiver from μ-blox is then analysed in section 7.2. Here the routing estimations are benchmarked against an existing layout and an alternative implementation (using CSPs instead of wirebond) is analyzed.

7.1. MP3 Player

Portable MP3 players have become famous during the last year. They consist of flash memories, a hardware decoder, a digital to analog converter and a microcontroller. The existing solutions are as large as a standard walkman featuring tapes. In fact, the smallest possible size on a PCB using SMT technology is 80 * 60 mm as shown in figure 7.1.

![Figure 7.1: Chip Limited Size of MP3 Player in SMT Technology: 80 * 60 mm](image)

HDP technology enables the designer to build a key holder sized player. In spite of the smaller outer size, more memory can be included because bare die attach requires only half the area of TSOP soldering.

Thus, the following case base assumptions are used:

- 128 MByte flash
- Maximum size 70 * 40 mm including case
The size constraints cannot be reached with an SMT solution because double sided assembly is impossible due to the fact that the back side of the board is required to hold the button contacts and to connect the LCD display.

7.1.1. Implementation

Figure 7.2 shows a mockup of a wirebonded implementation on an SBU substrate of 50 × 30 mm. It consists of four flash memories, the MAS MP3 decoder and the digital to analog converter (DAC), both from Intermetall, a microcontroller (µC) plus various passive components.

![Figure 7.2: MP3 Player Mockup Wirebonded onto an SBU Substrate: 50 × 30 mm (Courtesy of Art of Technology AG)](image)

The area required for the passive components is given from their size. They can be routed on this area because the interconnect requirements are very low. Therefore, the analysis is concentrated on the active components as summarized in Table 7.1. The parameters are minimum pad pitch ($C_{PP}$), chip length ($C_{Length}$), chip width ($C_{Width}$), number of on chip pads ($C_{N}$), number of on chip power pads ($C_{P}$) and the padlocation classification on the IC.

<table>
<thead>
<tr>
<th>Name</th>
<th>Count</th>
<th>$C_{PP}$ [µm]</th>
<th>$C_{Length}$ [µm]</th>
<th>$C_{Width}$ [µm]</th>
<th>$C_{Height}$ [µm]</th>
<th>$C_{N}$</th>
<th>$C_{P}$</th>
<th>Padlocations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash</td>
<td>4</td>
<td>430</td>
<td>14500</td>
<td>10800</td>
<td>400</td>
<td>70</td>
<td>4</td>
<td>short</td>
</tr>
<tr>
<td>MAS</td>
<td>1</td>
<td>300$^1$</td>
<td>5100</td>
<td>4480</td>
<td>400</td>
<td>46</td>
<td>8</td>
<td>all</td>
</tr>
<tr>
<td>DAC</td>
<td>1</td>
<td>300$^1$</td>
<td>3400</td>
<td>3340</td>
<td>400</td>
<td>35</td>
<td>7</td>
<td>all</td>
</tr>
<tr>
<td>µC</td>
<td>1</td>
<td>250</td>
<td>7000</td>
<td>6000</td>
<td>400</td>
<td>80</td>
<td>10</td>
<td>all</td>
</tr>
</tbody>
</table>

7.1.2. Substrate Technology

The player has low routing requirements and needs a substrate that allows double side assembly. These requirements can be fulfilled with a laminate technology but only with difficulties with other substrate technologies. Thus, for the implementation of the MP3 player a standard PCB and sequential build-up (SBU) were considered. Their designrules can be found in table 7.2. Here

$^1$Average chip pad pitch ($C_{PP}$) because their pad pitch is so irregular that the escaping requirements would be overestimated.
7.1. MP3 Player

$L_{EscVias}$ is the number of lines that pass between escaping through hole vias as explained in chapter 4. In the next subsection the footprint of the active components is calculated.

Table 7.2: Substrate Technologies for MP3 Player

<table>
<thead>
<tr>
<th>Technology</th>
<th>Line Width</th>
<th>Line Space</th>
<th>Via Land</th>
<th>Routing Pitch</th>
<th>$L_{EscVias}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCB</td>
<td>125</td>
<td>125</td>
<td>650</td>
<td>512</td>
<td>1</td>
</tr>
<tr>
<td>SBU</td>
<td>80</td>
<td>80</td>
<td>300</td>
<td>270</td>
<td>n/a</td>
</tr>
</tbody>
</table>

7.1.3. Bonding Options and Footprints

Neither the latest flash memories nor the Intermetall components are available in CSP package. Flip chip was considered as very difficult because the Intermetall dies have local pad pitches as low as 150 μm. Therefore wirebonding was selected as an alternative to SMT assembly. All components can be wirebonded with one pad row on the substrate. Their footprints calculated for SBU and PCB substrate technology (table 7.2) can be found in table 7.3. It can be seen that the footprints on the PCB are about twice as large as on the SBU substrate.

Table 7.3: Footprints for MP3 Player

<table>
<thead>
<tr>
<th>Component</th>
<th>$k$</th>
<th>$d_{Esc}$ [μm]</th>
<th>$O_{Esc}$ [μm]</th>
<th>Footprint Area [mm²]</th>
<th>$k$</th>
<th>$d_{Esc}$ [μm]</th>
<th>$O_{Esc}$ [μm]</th>
<th>Footprint Area [mm²]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash</td>
<td>5</td>
<td>957</td>
<td>105956</td>
<td>266</td>
<td>1</td>
<td>194</td>
<td>2420</td>
<td>175</td>
</tr>
<tr>
<td>MAS</td>
<td>3</td>
<td>1025</td>
<td>7400</td>
<td>148</td>
<td>2</td>
<td>2886</td>
<td>39</td>
<td>58</td>
</tr>
<tr>
<td>DAC</td>
<td>3</td>
<td>1025</td>
<td>7400</td>
<td>115</td>
<td>2</td>
<td>2886</td>
<td>39</td>
<td>58</td>
</tr>
<tr>
<td>μC</td>
<td>4</td>
<td>1025</td>
<td>9450</td>
<td>254</td>
<td>3</td>
<td>260</td>
<td>3460</td>
<td>98</td>
</tr>
</tbody>
</table>

Table 7.3 further summarizes the interim data such as the number of vias in an escaping row ($k$), the distance per via ($d_{Esc}$) and the escaping overhead ($O_{Esc}$). On the PCB substrate the calculation model for large k's (subsection 4.1.2.4) was used for all dies but the flash memories.

With these footprints, the chip limited size is derived from a placement in JavaCAD as described in the next subsection.

7.1.4. Chip Limited Size

The chip limited size for both options was derived from the edited placement in the JavaCAD. As shown in figure 7.3 it is 72 * 30 mm on a PCB, or 59 * 18 mm on an SBU substrate. This size covers only the active components. The passive components are an add-on of 666 mm² that enlarge the final substrate width.

The mockup in figure 7.2 is shorter than the chip limited size because the four flash dies are wider than the ones on the left lower side of the mockup.

7.1.5. Routing Limited Size

Seraphim’s estimation showed that the components can be routed on the chip limited size using two signal layers for both substrate technologies. As shown in table 7.4 the two signal layers are not used up to their maximum.
To check the above results, a global routing was performed on the channel graph in figure 7.4. The estimation showed that the SBU substrate can be routed on the chip limited size (table 7.5). For the PCB however a enlargement of 10% is estimated. The channel usage for the SBU substrate is shown in figure 7.5. The figure shows that the channel between the flash memories and the μC is the most congested. Thus, it was moved to the right when the global routing estimation demanded a spreading ($F > 1$).

### Table 7.4: Seraphim’s Estimation MP3

<table>
<thead>
<tr>
<th>Substrate</th>
<th>$P_D$ [mm]</th>
<th>$N_T$</th>
<th>$N_{Layer}$</th>
<th>Size [mm²]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCB</td>
<td>13</td>
<td>204</td>
<td>1.8</td>
<td>72*30</td>
</tr>
<tr>
<td>SBU</td>
<td>10</td>
<td>204</td>
<td>1.3</td>
<td>59*18</td>
</tr>
</tbody>
</table>

7.1.6. Final Selection

Both implementations (PCB and on SBU) are feasible, but for the PCB substrate an extra effort is required to reduce its length. The size reduction possible with the SBU substrate does not add
much to the cost (1.5%) because the components are quite expensive. The SBU technology allows to build the smallest player featuring 128 MB memory. In fact, SBU substrate technology allows to build the player within 65*35*15 mm including analog circuitry and package.

In the next section a high volume product, a GPS module, is analyzed.

### 7.2. GPS MCM

In 1997, µ-blox ag, an ETH Zürich spin-off company, launched its series of ultra-miniature global positioning system (GPS) receivers. The GPSMS1 is a fully self contained receiver multichip module (MCM) at a single operating voltage of 3.3 V. Based on the SiRFstar/LX chip set manufactured by SiRF Technology, Inc., the module provides complete GPS signal processing from antenna input to serial data output (see Fig. 7.6[1]).

It is built using wirebonded bare dies as described in the next subsection. The top side of the module defined the minimum size of the module. However, this size is given from the analog circuit size and does not depend on the routing. Thus, we analyzed the bottom side where a microprocessor, a DSP correlator, a flash memory and an SRAM are mounted.

During the last years, chip-scale packages (CSPs) have shown to be a viable alternative to bare dies due to their less problematic handling, better obtainability, and easier mounting procedure. Therefore, the models were also used to analyze and compare a CSP solution to the existing wire bonded implementation. A first attempt calculating the impact of CSPs for the GPS was done in [83]. In this section the findings are recalculated and corrected.
7.2.1. System Advantages and actual Implementation

The GPSMS1 fits into the form factor of a PLCC84 package (plastic leaded chip carrier). With its small size of 29.3×29.3 it has been the smallest fully self contained GPS available for two years now. Besides its smallness it relieves the system designer from challenging RF design and RF relevant production tolerances.

The PLCC84 package allows dual side assembly. On the top side, the RF front end chip, antenna connector, and additional components are mounted in surface-mount technology (partially housed), and on the bottom side a microprocessor, a DSP correlator, a flash memory and an SRAM are mounted as bare dies, wire bonded and overmolded.

The SBU (laminate) substrate technology used for the GPSMS1 features a standard laminate core with high-density sequential build-up layers (SBU), thus providing enough space for interconnection and separate analog/digital ground planes. Due to the extremely tight outer form factor (29.3 mm by 29.3 mm) and the high number of connections, leading-edge design rules are needed to provide the interconnect density required. Currently, 80/80/300 design rules are used.

7.2.2. Bonding Options and Footprints

In this subsection we compare the footprints of the existing wire bonded on SBU solution to alternatives featuring CSP or flip chip on laminate technologies as shown in table 7.6.

The bottom side consists of four dies or CSPs together with small SMT components and connectors. We neglect the SMT components and the connectors by reducing the available substrate area from 29.3 * 29.3 mm to 28*26 mm for the placement. The routing can use the whole area because the remaining components require hardly any routing.

7.2.2.1. Wirebond

Mounting dies with wire bond requires a significant area overhead to connect the ICs with the substrate. To calculate the footprints of the four dies, they are characterized with five parameters

---

Figure 7.6: Schematic of the μ-blox AG GPSMS1
7.2. GPS MCM

Table 7.6: Substrate Technologies for GPS MCM

<table>
<thead>
<tr>
<th>Technology</th>
<th>Line Width</th>
<th>Line Space</th>
<th>Via Land</th>
<th>Routing Pitch</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBU1</td>
<td>80</td>
<td>80</td>
<td>300</td>
<td>270</td>
</tr>
<tr>
<td>SBU2</td>
<td>100</td>
<td>100</td>
<td>350</td>
<td>325</td>
</tr>
</tbody>
</table>

as shown in table 7.7. These parameters are minimum pad pitch \( C_{PP} \), chip length \( C_{Length} \), chip width \( C_{Width} \), number of on chip pads \( C_N \) and the pad location classification on the IC.

Table 7.7: Wirebond data for footprint calculation

<table>
<thead>
<tr>
<th>Component</th>
<th>( C_{PP} )</th>
<th>( C_{Length} )</th>
<th>( C_{Width} )</th>
<th>( C_N )</th>
<th>Padlocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSP</td>
<td>80</td>
<td>3400</td>
<td>3400</td>
<td>100</td>
<td>all</td>
</tr>
<tr>
<td>( \mu P )</td>
<td>170</td>
<td>5400</td>
<td>5300</td>
<td>100</td>
<td>all</td>
</tr>
<tr>
<td>Flash</td>
<td>140</td>
<td>7090</td>
<td>3560</td>
<td>44</td>
<td>short</td>
</tr>
<tr>
<td>SRAM</td>
<td>150</td>
<td>4900</td>
<td>4500</td>
<td>59</td>
<td>long</td>
</tr>
</tbody>
</table>

Using the equations (4.65) to (4.80) in chapter 4, the footprint is calculated with the substrate parameters of SBU1 and the following wire bond parameters:

- Wire bond pad width \( B_{PWidth} \) 150 \( \mu m \)
- Wire bond pad length \( B_{PLength} \) 250 \( \mu m \)
- Minimum distance between IC and bondpads \( C_{2P} \) 500 \( \mu m \)

As shown in table 7.8, all components require two wire bond rows. These two rows add an overhead of about 4.6 mm to the sides where bond pads are located. The DSP has a larger overhead due to its \( k = 3 \) vias in one row for escaping which are required to escape its very small pitch. Due to this overhead the footprint area is double the chip’s size or even larger. These results can now be used to generate the chip limited size as described below.

Table 7.8: Wire bond footprint calculation

<table>
<thead>
<tr>
<th>Component</th>
<th>( N_{Row} )</th>
<th>( k )</th>
<th>( D_{Esc} ) ([\mu m])</th>
<th>( O_{Esc} ) ([\mu m])</th>
<th>( A_{Esc} ) ([mm^2])</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSP</td>
<td>2</td>
<td>3</td>
<td>287</td>
<td>6144</td>
<td>91</td>
</tr>
<tr>
<td>( \mu P )</td>
<td>2</td>
<td>2</td>
<td>169</td>
<td>4516</td>
<td>97</td>
</tr>
<tr>
<td>Flash</td>
<td>2</td>
<td>2</td>
<td>256</td>
<td>4864</td>
<td>42</td>
</tr>
<tr>
<td>SRAM</td>
<td>2</td>
<td>2</td>
<td>233</td>
<td>4772</td>
<td>45</td>
</tr>
</tbody>
</table>
7.2.2.2. CSP

To date, most of the SRAM and flash memories entering the market are also available in CSP packages. For the two remaining components, the microprocessor (\( \mu P \)) and the correlator (DSP), a custom CSP packaging is possible. They add an investment cost or non-recurring expenditure (NRE) share to the prize for a single CSP. However, CSPs have the potential to decrease the substrate requirements.

To evaluate this option the footprint is also calculated with the substrate SBU1, and the use of relaxed substrate rules of SBU2 is evaluated because it would allow substrate cost savings.

Table 7.9: CSP data for footprint calculation

<table>
<thead>
<tr>
<th></th>
<th>( \mu P )</th>
<th>DSP</th>
<th>Flash</th>
<th>SRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ball pitch [mm]</td>
<td>0.5</td>
<td>0.65</td>
<td>0.5</td>
<td>0.65</td>
</tr>
<tr>
<td>Pad size [mm]</td>
<td>0.3</td>
<td>0.32</td>
<td>0.3</td>
<td>0.32</td>
</tr>
<tr>
<td>No. I/Os whereas</td>
<td>100</td>
<td>89</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Pwr,Gnd, NC</td>
<td>22</td>
<td>11</td>
<td>48</td>
<td>39</td>
</tr>
<tr>
<td>Pad Area [mm²]</td>
<td>6.9*6.9</td>
<td>7.15*7.15</td>
<td>7.1*7.1</td>
<td>8.4*7.1</td>
</tr>
<tr>
<td>Outer CSP size [mm²]</td>
<td>8*8</td>
<td>8*8</td>
<td>7.25*7.25</td>
<td>8.8*7.5</td>
</tr>
</tbody>
</table>

The SRAM as well as the flash is available as CSP with 0.75 and 0.8 mm pad pitch, respectively. The remaining dies (\( \mu P \) and DSP) would have to be packaged in a custom package using a 0.5 or 0.65 mm pad pitch as summarized in table 7.9. With the equations (4.36) to (4.39) the footprint for CSPs are calculated. First the number of lines that can pass between the pads plus one (\( l_{top} \)) and the lines between two vias plus one (\( l_{in} \)) were calculated for the substrate technologies SBU1 and SBU2. Thus, \( l_{top} \) and \( l_{in} \) are the number of rows that can be escaped on the top respective inner layer. From them the number of layers were derived as summarized in table 7.10. Whereas on the SBU1 maximum two layers are necessary, three layers are needed to escape on SBU2 for the DSP with 0.5 mm pad pitch. The component footprint is in all options defined by the keepout size as marked bold in the table. Furthermore, it is the same size on both substrates for all components except the DSP. For the DSP the 0.65 mm pitch package is only 22% larger than the 0.5 mm package. Thus, if the chip limited size as described below is small enough for the larger solution, we will use it.

7.2.3. Chip Limited Size

With the above footprint calculations the minimum size can be derived from a trial placement using a large border on all sides to enable routing around as shown in figure 7.7. Note that both options have the same chip limited size (21.5*20 mm). Whereas the CSPs fit nicely together, the placement of the wire bonded dies show a lot of lost area. However, this area is required for routing as considered in the next section.

7.2.4. Routing Limited Size

The bottom side is routing limited on an SBU substrate as can be seen in figure 7.8a showing a first version of the receiver. A PCB substrate was not possible because the through hole vias
Table 7.10: Results of the footprint calculation

<table>
<thead>
<tr>
<th></th>
<th>μP 0.5mm</th>
<th>μP 0.65mm</th>
<th>DSP 0.5mm</th>
<th>DSP 0.65mm</th>
<th>Flash</th>
<th>SRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>(l_{\text{top}})</td>
<td>(80/80/300)</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>(100/100/350)</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(l_{\text{un}})</td>
<td>(80/80/300)</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>(100/100/350)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td># Layers req.</td>
<td>(80/80/300)</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(100/100/350)</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Area incl. Escaping (\text{[mm}^2)</td>
<td>(80/80/300)</td>
<td>7.7&quot;x7.7&quot;</td>
<td>8.1&quot;x8.1&quot;</td>
<td>7.1&quot;x7.1&quot;</td>
<td>8.4&quot;x7.1&quot;</td>
<td>6.8&quot;x5.2&quot;</td>
</tr>
<tr>
<td></td>
<td>(100/100/350)</td>
<td>8&quot;x8</td>
<td>8.85&quot;x8.85</td>
<td>8.25&quot;x8.25</td>
<td>9.5&quot;x8.5</td>
<td>7.5&quot;x5.9</td>
</tr>
<tr>
<td>Keypout Area (\text{[mm}^2)</td>
<td>9&quot;x9</td>
<td>9&quot;x9</td>
<td>8.25&quot;x8.25</td>
<td>9.8&quot;x8.5</td>
<td>10&quot;x7</td>
<td>9&quot;x7</td>
</tr>
</tbody>
</table>

![Diagram a: Wire bonding on SBU1, 21.5 mm x 20 mm](image1)
![Diagram b: CSPs with minimum pitch 0.65 mm on SBU2, 21.5 mm x 20 mm](image2)

Figure 7.7: Chip Limited Size of GPS MCM

would have guided the noise from the digital part into the analog receiver. Of the module size of 29.3*29.3 mm only 28*26 mm are used for the above dies in the actual solution on two routing layers.

In this subsection the routing estimations are benchmarked against the existing solution and they are used to estimate the routing requirements for the CSP alternative. First the routing requirements are estimated with Seraphim and then a detailed analysis is performed with global routing.

### 7.2.4.1. Seraphim

Seraphim’s estimation mainly depends on:

- Number of I/Os \(N_T\) = 206
- the chip pitch \(P_D\) = 9.5 mm derived from the placement in figure 7.7
- and the routing pitch \(P_W\) = 270 \(\mu m\) for SBU1 and 325 \(\mu m\) for SBU2
Equation (5.3) calculates for the above parameter values a size of 27*29 mm for two layers of SBU1. For SBU2 the layer count at this size is still 2.5. To reduce it to 2 layers the substrate is estimated to become 35*35 mm as shown in figure 7.9.

Seraphim's estimation is the same for both assembly methods. However, as can be seen in the placement in figure 7.7, wirebonding provides more space between the components. The difference is quantified in the more accurate estimation described below.
7.2.4.2. Global Routing Estimation

The global routing estimation was performed on the channel graphs shown in figure 7.10. It can be seen that the wirebonded solution has much more space between the components. However, in CSP package all components but the DSP processor require only one layer for escaping instead of two for wirebonding.

![Figure 7.10: Routing Channels for GPS MCM](image)

The global routing estimation showed that the area blocked by the CSPs cannot be compensated by their slightly reduced layer usage. As summarized in table 7.11 the necessary routing area is at least 10% larger for the CSP solution. Nevertheless, it should be possible to interconnect the CSPs on two layers using the SBU1 substrate. If three layers are available, then both solutions can be routed on a SBU2 substrate. However, it is not possible to interconnect the components on a two layer SBU2 substrate even with wirebonded dies.

<table>
<thead>
<tr>
<th>Substrate</th>
<th>Assembly</th>
<th>$N_{layer}$ (Signal)</th>
<th>$E$</th>
<th>Estimated Size [$mm^2$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBU1</td>
<td>Wirebond</td>
<td>2</td>
<td>1.32</td>
<td>28.4 * 26.4</td>
</tr>
<tr>
<td></td>
<td>CSP</td>
<td></td>
<td>1.48</td>
<td>30.3 * 30.3</td>
</tr>
<tr>
<td></td>
<td>Wirebond</td>
<td>3</td>
<td>1.11</td>
<td>23.8 * 22.2</td>
</tr>
<tr>
<td></td>
<td>CSP</td>
<td></td>
<td>1.29</td>
<td>28 * 25</td>
</tr>
<tr>
<td>SBU2</td>
<td>Wirebond</td>
<td>2</td>
<td>1.55</td>
<td>31.7 * 31.7</td>
</tr>
<tr>
<td></td>
<td>CSP</td>
<td></td>
<td>1.62</td>
<td>33.2 * 33.2</td>
</tr>
<tr>
<td></td>
<td>Wirebond</td>
<td>3</td>
<td>1.30</td>
<td>28.2 * 25.2</td>
</tr>
<tr>
<td></td>
<td>CSP</td>
<td></td>
<td>1.46</td>
<td>29.9 * 29.9</td>
</tr>
</tbody>
</table>
Table 7.12: Partial data for the cost/yield calculation

<table>
<thead>
<tr>
<th></th>
<th>wire bond 80/80/300 direct cost</th>
<th>yield</th>
<th>NRE</th>
<th>CSP 0.65mm direct cost</th>
<th>yield</th>
<th>NRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSP</td>
<td>8.50</td>
<td>0.999</td>
<td></td>
<td>10.50</td>
<td>0.999</td>
<td>3Sk</td>
</tr>
<tr>
<td>µP</td>
<td>5.75</td>
<td>0.999</td>
<td></td>
<td>7.75</td>
<td>0.999</td>
<td>3Sk</td>
</tr>
<tr>
<td>SRAM + Flash</td>
<td>6.50</td>
<td>0.998</td>
<td></td>
<td>7.15</td>
<td>0.998</td>
<td></td>
</tr>
<tr>
<td>Substrate</td>
<td>4.25</td>
<td>0.999</td>
<td></td>
<td>4.25</td>
<td>0.999</td>
<td></td>
</tr>
<tr>
<td>Assembly Back side</td>
<td>18.20</td>
<td>0.999</td>
<td>1</td>
<td>12.90</td>
<td>0.999</td>
<td></td>
</tr>
<tr>
<td>† without wire bond</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wire bond</td>
<td></td>
<td>0.999999999</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7.2.4.3. Comparison of the routing estimation methods

The calculation for two routing layers resulted in 29*27 mm using Seraphim’s approach and 28.5*26.5 mm with global routing as compared in figure 7.9. Here the result of Seraphim is plotted as curves. The lower is for SBU1 and the upper for SBU2. The global routing result is marked as bullets in the corresponding color. The bullets can be found on the two layers’ and the three layers’ line. The global routing estimation corresponds quite well with the existing layout. But Seraphim overestimates the area because 50% of the nets are busses with a shorter wire length than estimated. For the estimation with three layers this effect is overcompensated because Seraphim’s estimation does not take into account that for uneven layer count the wires in x and y direction can block each other. In fact Seraphim’s estimation showed a 30% smaller area for the wirebonded solution and a 50% smaller area for the CSP alternative than the global routing estimation.

The routing estimation showed that only two options are feasible: the existing implementation and a CSP alternative on a two layer SBU1 board. The calculated size and layer count is now used to perform a cost/yield analysis for the feasible options as described below.

7.2.5. Cost/yield analysis

After the technical considerations two options, the existing implementation and a CSP alternative on a two layer SBU1 board, are left. To select one of them, a cost calculation is required. The cost calculation was done with the ETH in-house tool MOE (Modular Optimization Environment) [57]. It features a graphical process modelling, allowing for an easy perception of the cost model structure.

In table 7.12, part of the underlying cost data can be found. The data detail the main differences between the existing and the new implementation. The main cost advantage of the CSP solution is caused by the lower backside assembly cost (by saving wire bonding and bond testing) and by the lower yield loss. On the other hand, the component cost is higher, and some investments for custom CSPs have to be incurred. All the cost/yield data for front side assembly, capping, final test and substrate remain unchanged.

The cost calculation results for a depreciation period of 2 years and a target quantity of 100'000 units per year can be found in Fig. 7.11. The cost savings of the CSP solution are in total about 3.5%. The direct cost reduction contributes around 1.5%, but the NRE is about 0.7% higher. The reduced yield loss has the highest contribution to the improvement.
7.2.6. Conclusion

The footprint analysis showed that the use of CSPs does not lead to further form factor reduction. In fact, with CSPs a 30% larger footprint is required. While for the memory components the entire CSP footprint equals almost the die size, saving some of the wire bond area, the microprocessor and DSP CSPs require more space. The much relieved I/O pitch of the CSPs would allow to reduce the overall design rules of the MCM substrate to 100/100/350. Unfortunately, the performed routing estimations showed that it is not possible to interconnect the components on the coarser substrate within the size constraints. The lower assembly cost (no wire bond) and the yield improvement from 96.5 to 99.5% lead to an overall module cost reduction of about 3.5%. This can be achieved although investments have to be made to obtain custom CSPs for the DSP and the μP. In fact, the cost evaluation revealed that the higher investment cost can be compensated by reduced yield loss.

Thus, the cost savings justify considering a CSP packaging solution for a next redesign of μ-blox' GPS MCM.

7.3. Summary

The two presented case studies showed the usage and the usefulness of the methods and models. As evaluated, the HDP technologies enable a very small MP3 player without compromises in play time. The GPS MCM is about three times smaller than an SMT alternative. The analysis showed that CSPs would help to reduce the cost. However, the cost reduction is too low to justify a redesign before it is not forced because of component changes. Both examples showed that Seraphim's estimation is too optimistic. To run a router to check the sizes requires days. Furthermore, a netlist, component definition and a placement in a layout

Figure 7.11: Results of the cost calculation, combination of MOE result windows; depreciation period 2 years, target quantity 100'000 units per year
tool is required thus forcing another effort. This time consuming tasks can be avoided by using the global routing estimation. It provides a size estimation and additionally shows congested areas.

The findings in this thesis are summarized in the last chapter and an outlook to further developments in technology and methods are given.
Summary and Conclusions

A methodology for MCM design including system analysis was presented. It can be utilized for systems using standard and/or custom ICs. The proposed design methodology manages the complexity and the many degrees of freedom HDP technologies offer. The developed design space exploration is summarized below.

To make this task tractable the following models were developed:

- Effective wiring pitch
- Number of layers vs. substrate side length depending on generic substrate types
- Footprint of components with any bonding and on any substrate technology
- Novel routing estimation

These models are summarized in the first two sections of this chapter. Furthermore, they were implemented as a diploma thesis at the Electronic Laboratory of ETH Zurich as summarized in section 8.3 "JavaCAD".

After the explanation of the very few limitations in section 8.4, the impact of the foreseen technology development is considered in section 8.5. This framework concludes with proposing further developments in section 8.6.

8.1. Design Space Exploration

A design space exploration is performed to synthesize a specification into a product. To manage the many degrees of freedom the technologies offer, a tree of possible build-ups is created as shown in figure 3.3. It consist of the steps

- Partitioning,
- Technology Selection,
- Design for Test,
- and concludes with a cost performance trade-off that enables a designer to choose ONE of them for implementation.

This thesis mainly covers the technology selection where

- the component footprints are calculated,
• the chip limited size derived from the footprints and
• the routing limited size are calculated

to extract the required parameters for the cost-performance trade-off. To do so, the models summarized in the next section were developed.

8.2. Substrate Area and Layer Count Estimation

A successful solution for estimating substrate area and layer count was provided. The footprint calculation together with a trial placement allows to estimate the minimum substrate size and layer count. The models are substrate technology independent. They are purely geometrical. Thus, the difference between the technologies is parameterized. The necessary routing resources can then very accurately be modelled with the presented global routing approach. The global routing objective, the minimization of the substrate area for a given number of layers, is reached by maximizing the channel usage under the components and minimizing the spreading of the channels between the components. This approach allows not only the minimization of substrate size but can easily be extended to additionally fulfil other constraints such as electrical performance [84]. The estimation is much more accurate than overall estimations such as Seraphim’s approach, which can underestimate the required wiring resources by more than 20% (subsection 5.6.4). This is due to the fact that the global routing estimation includes local congestions and calculates the required wiring length instead of estimating it heuristically. Therefore, it requires no safety margin. The presented results and other studies showed that it provides an accuracy better than ±5%. However, the new approach is not only more accurate but also needs more input and more computing power. But, compared to a trial and error approach by means of an autorouter, the input and computing power required is still low. Furthermore, the new approach calculates how large a substrate must be and how large spacing between the components is required, instead of “go/no go result” from a trial routing.

The models are purely based on geometrical data such as design rules, pad pitches and sizes. They do not utilize any build-up specific data. Therefore our approach is not limited to one substrate technology, but enables a comparison of several build-up alternatives. This also applies for the first level interconnect where not only flip chip is supported but also wire bond, TAB and even packaged components are included in our approach.

8.3. JavaCAD

To ease the usage of the method, as well as to have the models implemented in software, a tool called JavaCAD was written. JavaCAD allows to describe a system with very few parameters such as component size, number of I/Os, pad pitch, power dissipation and available packaging forms. Providing input of all the components in the project window, the other required data such as design rules, assembly rules etc. are taken from libraries. From these parameters a placement is generated. The placement can then be edited. From this placement as well as from the sum of area, Seraphim’s estimation provides a first hint of the layer count required for interconnection.

By providing a simplified netlist, the tool performs the developed global routing and calculates very accurate data such as substrate size and routing layer count. JavaCAD further includes a cost estimation and provides an ASCII interface to and from layout tools. Thus, it closes the chain for automated layout.
8.4. LIMITATIONS

The footprint calculation has no limitation within the provided models for the assembly technology. However, as soon as new assembly technologies are developed, one of the models will have to be adapted to them. Furthermore, the footprint estimation may overestimate if the pads of a component can be connected one-to-one to the neighbor, thus requiring no vias.

All routing estimations presented in this thesis are based on a single average routing pitch for all routing layers. This is not always true, but the channel capacity calculation in section 5.4.4 can be adapted to several routing pitches.

Usually, the presented routing pitch is overestimation. However, for through hole vias and more than six substrate layers it might be too optimistic as the through holes block all routing layers i.e. the substrate looks like an emmentaler cheese. This is easily avoided by replacing them with microvias.

8.5. Impact of the NTRS Roadmap on Substrate and Assembly Technologies

The NTRS roadmap [66] foresees a tremendous growth in chip pad count as shown in table 8.1. This pad count has to be managed by the assembly methods as summarized in table 8.2. The foreseen requirements for wirebond and TAB most probably will be managed, but they limit the maximum pad count per IC [85]. The very high pad count IC can only be assembled using Flip Chip bonding. The foreseen requirement of 50 micron pitch in year 2012 is very aggressive. Furthermore, the pitch will not only be limited by the manufacturable pad pitch but also by the escape routing on the substrate. The very small pitch not only requires finest wiring pitch but also many layers to escape and smallest vias that fit into the substrate pads or between.

The impact of these requirements can be analyzed with the models presented in chapter 4 because in spite of the foreseen tremendous development the presented models remain valid. However, the projected very small pad pitch might be escaped on a package substrate, but it will be too expensive to use the highest density technology for a module or board. Therefore, the foreseen ICs pads are redistributed to chip size packages. Their development was also analyzed in the NTRS
Table 8.3: CSP Needs [66] and calculated

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NTRS</td>
<td>250 nm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSP Pad Pitch [mm]</td>
<td>0.5</td>
<td>0.4</td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
<td>0.25</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>Pad Size [μm]</td>
<td>200</td>
<td>170</td>
<td>170</td>
<td>150</td>
<td>150</td>
<td>130</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Line Width [μm]</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Line Space [μm]</td>
<td>65</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>Calculated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Soldermask accuracy [μm]</td>
<td>55</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Line Width [μm]</td>
<td>80</td>
<td>70</td>
<td>70</td>
<td>70</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>30</td>
</tr>
<tr>
<td>Via Land Diameter [μm]</td>
<td>350</td>
<td>250</td>
<td>250</td>
<td>190</td>
<td>190</td>
<td>170</td>
<td>170</td>
<td></td>
</tr>
</tbody>
</table>

The roadmap shown in the upper part of Table 8.3. However, in the NTRS it is not considered that CSP pads also require a soldermask to avoid solder bridging. Therefore, the calculated rules for two lines between the pads enabling the escaping of three rows per layer is too optimistic as explained below.

The area consumed by the soldermask depends on its placement accuracy. To ensure that no soldermask is on the pad, the soldermask opening is widened by its accuracy on all sides. Therefore, the soldermask accuracy consumes four times its width between two pads. However, as two times the accuracy is usually larger than the required line space, the line spacing for the first line is already fulfilled. Based on these constraints the lower part of Table 8.3 was then calculated. The values for the soldermask accuracy are own projections into the future. It can be seen that only two rows are escaped at a reasonable pitch. The calculation of the maximum via land diameter showed feasible land sizes. The impact on the layer count was not considered because it depends on the row configuration and the power pins. JavaCAD can be used to analyze it.

However, the escaping requirements for a high performance component might be defined by local requirements such as a fast memory bus in one corner of an processor IC. To analyze its impact, the equations in chapter 4 are used.

Beside the analysis of IC and packaging requirements, the developed models are also the basis for further developments as described in the next section.

8.6. Future Work

The presented methods and models allow to generate a tree of feasible build-ups and to analyze them. However, not every step is covered with models. A most promising enhancement would be a thermal estimation that calculates the required thermal inserts. These inserts are then considered in the routing estimation.

At present, a tree of feasible build-ups for a module has to be generated manually by selecting the possible first level interconnect and substrate technologies. Based on the presented models the selection tree could be built automatically and the selection could be performed by a multiobjective optimization [81].

The global routing generates a plan how to route the interconnecting nets. For exact track and via positions and layer assignment a detailed router is required. At present, commercial PCB
routers are not capable to work with this planning. IC routers could be used but they are not
developed to work with via land diameters larger than the line width. Therefore, a router should
be developed that works with the provided planning and minimizes via usage. There were already
some activities as summarized in [86] and [87].

In spite of the proposed enhancements, the methods and models are ready for use and are
already implemented in software.
A

Utilized Parameters and their Symbols

In this appendix the parameters are summarized. It is suited as reference for the used symbols. Furthermore, it provides typical values for them. The appendix is organized in:

• Chip Parameters,
• Assembly Parameters,
• Substrate Parameters,
• and Calculated Parameters.
# Appendix A: Utilized Parameters and their Symbols

## A.1. Chip/Package Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Unit</th>
<th>typical</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{length}$</td>
<td>Chip length</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$C_{width}$</td>
<td>Chip width</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$C_{height}$</td>
<td>Chip height</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$C_{PP}$</td>
<td>Minimum pad pitch of a chip</td>
<td>micron</td>
<td>50 to 400 $\mu$m</td>
</tr>
<tr>
<td>$C_{DP}$</td>
<td>Average pad pitch of a chip</td>
<td>micron</td>
<td>50 to 800 $\mu$m</td>
</tr>
<tr>
<td>$C_{N}$</td>
<td>Number of chip pads</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C_{IO}$</td>
<td>Number of I/Os of a chip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$P_{2C}$</td>
<td>Chip pad center to chip border</td>
<td>micron</td>
<td>200 $\mu$m</td>
</tr>
<tr>
<td>$P_{2S}$</td>
<td>Chip pad center to chip border on left and right side</td>
<td>micron</td>
<td>300 to 600 $\mu$m</td>
</tr>
<tr>
<td>$P_{package width}$</td>
<td>Package width</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$P_{package length}$</td>
<td>Package length</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$P_{Pitch}$</td>
<td>Pad Pitch usually for packages</td>
<td>micron</td>
<td>300 to 2540 $\mu$m</td>
</tr>
<tr>
<td>$P_{Width}$</td>
<td>Pad width</td>
<td>micron</td>
<td>30 to 800 $\mu$m</td>
</tr>
<tr>
<td>$P_{Length}$</td>
<td>Pad length</td>
<td>micron</td>
<td>30 to 800 $\mu$m</td>
</tr>
<tr>
<td>$N_{1}$</td>
<td>Number of area array rows on longer side</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N_{2}$</td>
<td>Number of area array rows on smaller side</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N_{BE}$</td>
<td>Number missing ball rows in the center of an area array on longer side</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N_{BE}$</td>
<td>Number missing ball rows in the center of an area array on smaller side</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N_{1I}$</td>
<td>Number ball rows in the center of an empty center on longer side</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N_{1I}$</td>
<td>Number ball rows in the center of an empty center on smaller side</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A.2. Assembly Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Unit</th>
<th>Typical</th>
</tr>
</thead>
<tbody>
<tr>
<td>$BP_{Length}$</td>
<td>Substrate bond pad length</td>
<td>micron</td>
<td>$250 \mu m$</td>
</tr>
<tr>
<td>$BP_{Width}$</td>
<td>Substrate bond pad width</td>
<td>micron</td>
<td>$100 - 200 \mu m$</td>
</tr>
<tr>
<td>$C2P_{min}$</td>
<td>Minimum distance between chip and any metal (gluing)</td>
<td>micron</td>
<td>$500 \mu m$</td>
</tr>
<tr>
<td>$H_{loop}$</td>
<td>Wire bond loop height</td>
<td>micron</td>
<td>$150 \mu m$</td>
</tr>
<tr>
<td>$L_{H_{in}}$</td>
<td>Minimum horizontal wire bond length</td>
<td>micron</td>
<td>$600 \mu m$</td>
</tr>
<tr>
<td>$L_{H_{max}}$</td>
<td>Maximum horizontal wire bond length</td>
<td>micron</td>
<td>$2000 \mu m$</td>
</tr>
<tr>
<td>$\beta_{CP}$</td>
<td>Maximum on-chip angle of wirebond</td>
<td>degree</td>
<td>$45^\circ$</td>
</tr>
<tr>
<td>$\Delta \delta_{BP}$</td>
<td>Maximum substrate bondpad to wirebond misalignment</td>
<td>degree</td>
<td>$10^\circ$</td>
</tr>
<tr>
<td>$B_{Bp-spacing}$</td>
<td>Bond pad to bond pad spacing of adjacent ICs</td>
<td>micron</td>
<td>$400 \mu m$</td>
</tr>
<tr>
<td>$P_{ODLR}$</td>
<td>Pitch for outer lead bonding of TAB</td>
<td>micron</td>
<td>$700 - 300 \mu m$</td>
</tr>
<tr>
<td>$C2P_{TAB}$</td>
<td>Minimum distance between chip and TAB pads</td>
<td>micron</td>
<td>$700 \mu m$</td>
</tr>
<tr>
<td>$L_{P_{Length}}$</td>
<td>Pad length</td>
<td>micron</td>
<td>$30$ to $800 \mu m$</td>
</tr>
</tbody>
</table>

A.3. Substrate Parameters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Range/Unit</th>
<th>Typical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ll</td>
<td>Type</td>
<td>PCB, mcm, hybrid, lice, lice photoe, mcmD, DonL</td>
<td></td>
</tr>
<tr>
<td>$L_{Width}$</td>
<td>Line width</td>
<td>micron</td>
<td>$10$ to $300 \mu m$</td>
</tr>
<tr>
<td>$L_{Space}$</td>
<td>Line spacing</td>
<td>micron</td>
<td>$15$ to $400 \mu m$</td>
</tr>
<tr>
<td>$L_{Pitch}$</td>
<td>Line pitch (line spacing + line width)</td>
<td>micron</td>
<td>$25$ to $700 \mu m$</td>
</tr>
<tr>
<td>$D_{ViaLand}$</td>
<td>Via land diameter</td>
<td>micron</td>
<td>$20$ to $800 \mu m$</td>
</tr>
<tr>
<td>$L_{Vias}$</td>
<td>Number of lines between vias for routing</td>
<td></td>
<td>$3$</td>
</tr>
<tr>
<td>$L_{C0Vias}$</td>
<td>Number of lines between through hole vias for escaping</td>
<td></td>
<td>$1$</td>
</tr>
<tr>
<td>$BP_{Width}$</td>
<td>Minimum bondpad width</td>
<td>micron</td>
<td>$100$ to $200 \mu m$</td>
</tr>
<tr>
<td>$C2D_{Substrate}$</td>
<td>Chip to Bondpad</td>
<td>micron</td>
<td>$500 \mu m$</td>
</tr>
<tr>
<td>$t$</td>
<td>Line thickness</td>
<td>micron</td>
<td>$2$ to $30 \mu m$</td>
</tr>
<tr>
<td>$h$</td>
<td>Dielectric thickness</td>
<td>micron</td>
<td>$5$ to $600 \mu m$</td>
</tr>
<tr>
<td>$\varepsilon_r$</td>
<td>Dielectric permittivity</td>
<td></td>
<td>$2.65$ to $12$</td>
</tr>
</tbody>
</table>
### Appendix A: Utilized Parameters and their Symbols

#### A.4. Calculated Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Unit</th>
<th>typical range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{pads}$</td>
<td>Size governed by a component's pads</td>
<td>$\mu m^2$</td>
<td></td>
</tr>
<tr>
<td>$A_{Esc}$</td>
<td>Component footprint if defined from escaping</td>
<td>$\mu m^2$</td>
<td></td>
</tr>
<tr>
<td>$d_{Via}$</td>
<td>Distance from via to via in a row for escaping</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$k$</td>
<td>Via count in a row for escaping of a component</td>
<td></td>
<td>1 to 10</td>
</tr>
<tr>
<td>$d_{Esc}$</td>
<td>Distance an escaping via enlarges the escaping overhead</td>
<td>micron</td>
<td>$0$ to $D_{Via} + D_{Space}$</td>
</tr>
<tr>
<td>$N_{Esc}$</td>
<td>Layer count for escaping of a component</td>
<td></td>
<td>1 to 4</td>
</tr>
<tr>
<td>$N_{Row}$</td>
<td>Number of wire bond rows</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B2B$</td>
<td>Distance from wirebond row to wirebond row</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$P_{D}$</td>
<td>Chip pitch</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$N_{T}$</td>
<td>Sum of all component's I/Os</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L$</td>
<td>Total wiring length</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$A_{W0}$</td>
<td>Chip limited area for Seraphim's estimation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{0}$</td>
<td>Total wiring length estimated for $A_{W0}$</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$P_{r}$</td>
<td>Routing pitch</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$N_{Layer}$</td>
<td>Number of signal layers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$O_{Esc}$</td>
<td>Escaping overhead</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$O_{Esc2}$</td>
<td>Part of the escaping overhead that cannot be used by the global routing channel between two components</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$L_{pad}$</td>
<td>Number of lines passing between two bumps or batts on the top layer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{via}$</td>
<td>Number of lines passing between two vias with ball/bump pitch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{in}$</td>
<td>Number of rows of an area array that are escaped on an inner layer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{top}$</td>
<td>Number of rows of an area array that are escaped on the top layer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$W_{n}$</td>
<td>Width of a via row for special escaping</td>
<td>micron</td>
<td></td>
</tr>
<tr>
<td>$E_{c}$</td>
<td>Number of lines in a via row for special escaping</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$V_{c}$</td>
<td>Via count in a via row for special escaping</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bibliography


[60] Intel Corporation, Pentium Processors with Voltage Reduction Technology (Smartdie Product specification), 1996.


BIBLIOGRAPHY


