Yangdong Deng
Wojciech P. Maly

3-Dimensional VLSI

A 2.5-Dimensional Integration Scheme

TSINGHUA UNIVERSITY PRESS
Springer

Yangdong Deng
Wojciech P. Maly

3-Dimensional VLSI

A 2.5-Dimensional Integration Scheme

Yangdong Deng
Wojciech P. Maly

3-Dimensional VLSI

A 2.5-Dimensional Integration Scheme

With 63 figures

TSINGHUA UNIVERSITY PRESS
Springer

Authors

Prof. Yangdong Deng
Institute of Microelectronics
Tsinghua University
Beijing 100084, P. R. China
Email: dengyd@tsinghua.edu.cn

Prof. Wojciech P. Maly
Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891
USA
Email: maly@ece.cmu.edu

ISBN 978-7-302-21165-5
Tsinghua University Press, Beijing

ISBN 978-3-642-04156-3 e-ISBN 978-3-642-04157-0
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2009933256

© Tsinghua University Press, Beijing and Springer-Verlag Berlin Heidelberg 2010
This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.
The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: Frido Steinen-Broo, EStudio Calamar, Spain

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Yangdong Deng
Wojciech P. Maly

3 维超大规模集成电路——2.5 维集成方案

3-Dimensional VLSI — A 2.5-Dimensional Integration Scheme

With 63 figures and 16 tables

TSINGHUA UNIVERSITY PRESS
Springer

内 容 简 介

本书提出一种新的 3 维超大规模电路集成方案，即 2.5 维集成。根据这一集成方案实现的电子系统将由多层单片集成芯片叠加而成，芯片间将由极细小尺度的“垂直联线”实现电气连接。这一新集成方案能够在很大程度上克服积累成品率损失的问题。

本书从制造成本和设计系统性能两方面探讨 2.5 维集成的可行性。首先，作者建立了一个成本分析模型来比较各种典型集成方案，分析数据表明 2.5 维集成具备制造成本上的优越性。从设计性能角度，作者完成了全定制和专用集成电路两种设计风格的一系列设计实例研究，从而证明了 2.5 维集成能够实现传统单片集成不能达到的系统性能。同时，为了实现 2.5/3 维集成电路版图，作者也开发了第一代 2.5 维/3 维物理设计 EDA 工具。

本书适合集成电路工艺开发人员和决策人士、集成电路设计人员、电子设计自动化研发人员和决策人士参考。

版权所有，侵权必究。侵权举报电话：010-62782989　13701121933

图书在版编目(CIP)数据

3 维超大规模集成电路：2.5 维集成方案 = 3-Dimensional VLSI: A 2.5-Dimensional Integration Scheme: 英文 / 邓仰东，(美) 马利 (Maly, W. P.) 著. —北京：清华大学出版社，2009.12
ISBN 978-7-302-21165-5

Ⅰ.3… Ⅱ.①邓… ②马… Ⅲ.超大规模集成电路－英文 Ⅳ.TN47

中国版本图书馆 CIP 数据核字(2009)第 180065 号

责任编辑：陈志辉
责任校对：王淑云
责任印制：

出版发行：清华大学出版社　　**地　　址：**北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者：
经　　销：全国新华书店
开　　本：153×235　**印张：**12.75　**彩插：**2　**字数：**175 千字
版　　次：2009 年 12 月第 1 版　**印次：**2009 年 12 月第 1 次印刷
印　　数：1～0000
定　　价：00.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010-62770177 转 3103　产品编号：029613-01

Preface

Today we are seeing strong demand for integrating more functionality onto silicon. Nonetheless, we are soon approaching the limit of Moore's law. In fact, the fundamental physics laws preclude the scaling of CMOS devices below a certain dimension. On the other hand, so far no alternative technologies are likely to mature and replace CMOS in the coming 15 years. Then how could the semiconductor industry continue to provide integration capacity for constantly increasing functionality?

3-D integration is a natural solution to address the above problems. Orthogonal to shrinking feature size, a 3-D integrated VLSI system would deploy multiple device layers to improve integration density. Moreover, since the vertical inter-chip interconnects could provide a shortcut to break long signal paths, a 3-D IC would have opportunity for improved circuit performance. Inspired by the great potential, many 3-D integration schemes and fabrication technologies have been proposed in the last a few years.

As pioneers in this new 3-D arena, the authors of this book designed a new 3-D integration scheme, so-called 2.5-D integration. According to this concept, a VLSI system is built as a 3-dimentional assembling of monolithic chips with small-scaled inter-chip interconnections. With a carefully designed, incremental and hierarchical testing methodology, this approach would largely overcome the accumulative yield loss problem hindering other 3-D integration schemes.

In this book, the authors evaluated the feasibility of the 2.5-D integration from both cost and performance perspectives. They established an analytical cost model to compare the manufacturing cost of different VLSI integration styles. The cost analysis shows that the 2.5-D scheme could offer significant cost saving over other schemes. Secondly, the authors performed design case studies on real-world designs. These studies demonstrate the strong potential of 2.5-D integrated designs for higher performance. To study the characteristics of 3-D layouts, the authors constructed a prototype EDA tool-chain consisting of 2.5/3-D floorplanning, placement, and routing tools. With these tools, a synthesized netlist could be automatically implemented as manufacturable layout.

To the best of my knowledge, this book is the first one to give a complete overview of the 3-D integration problem. It would provide valuable information for readers from various communities, such as semiconductor fabrication process developers, IC designers, and EDA R&D practitioners. The book could also serve as an excellent reference for graduates majoring in microelectronics.

Prof. Zhihua Wang
Professor
Institute of Microelectronics
Tsinghua University
Beijing, China

November, 2008

Acknowledgements

This book is based on the first author's Ph.D. work. So we would like to thank his Ph.D. committee members, Dr. Wilfred Haensch, Prof. Larry Pileggi, Prof. Radu Marculescu, and Dr. Herman Schmit, for reviewing this research work and providing valuable feedback. We are extremely grateful to Dr. Herman Schmit and Dr. David Whelihan, who kindly provide us with the complete design data of PipeRench. During the whole period of this research, we have been discussing with many other researchers and their opinions have greatly improved the quality of this work. Here we would like to show our gratitude to Prof. Rob A. Rutenbar, Prof. Qiao Lin, Dr. P. K. Nag, Prof. Sung-Kyu Lim, Prof. Yuan Xie, Prof Peng Li, Dr. Jingcao Hu, Prof. Chunsheng Liu, Julia Fei, Tao Lin, Dr. Thomas Zanon, Dr. Yi Wang, and many others. The authors are also thankful for Prof. Zhihua Wang for writing the preface.

Contents

List of Figures and Tables

1 Introduction

Yangdong (Steve) Deng

Institute of Microelectronics, Tsinghua University
Beijing 100084, P. R. China, dengyd@tsinghua.edu.cn

Wojciech P. Maly

Electrical and Computer Engineering Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891 USA, maly@ece.cmu.edu

Abstract In this chapter we elaborate on the need for new 3-dimensional VLSI paradigms by extrapolating the trend of technology development. On such a basis, we will propose our target 2.5-D integration scheme, and then explain its advantages. The fabrication, testing, and design technologies to enable the 2.5-D scheme are explained. Finally we are going to introduce the objectives and organization of this book.

Keywords 3-dimensional VLSI, 2.5-D integration, inter-chip contact, inter-connection, fabrication, test, design technology.

The semiconductor industry has been and will continue to be driven by the consumer demands for superior performance and functionality. To keep pace with

such demands, it is essential to maintain the momentum of shrinking process feature size so as to pack more devices on a single silicon die. As a matter of fact, the complexity of the integrated circuit (IC) system has always been growing at the speed delineated by the Moore's Law since the invention of the first integrated circuit. From the beginning of the 1990s, the speed of increasing complexity has even been accelerated with the introduction of broadband and multimedia applications. One such exemplar application is illustrated in Fig. 1.1, where each dot representing the number of gates on a given generation of NVidia's flagship graphic processing unit (GPU)[1]. The dotted line indicates the number of gates predicted by the Moore's Law. Clearly, the GPU chips would integrate a greater number of transistors than that predicted by the Moore's Law. Similar trends could be observed in other applications domains like wireless chipsets[2].

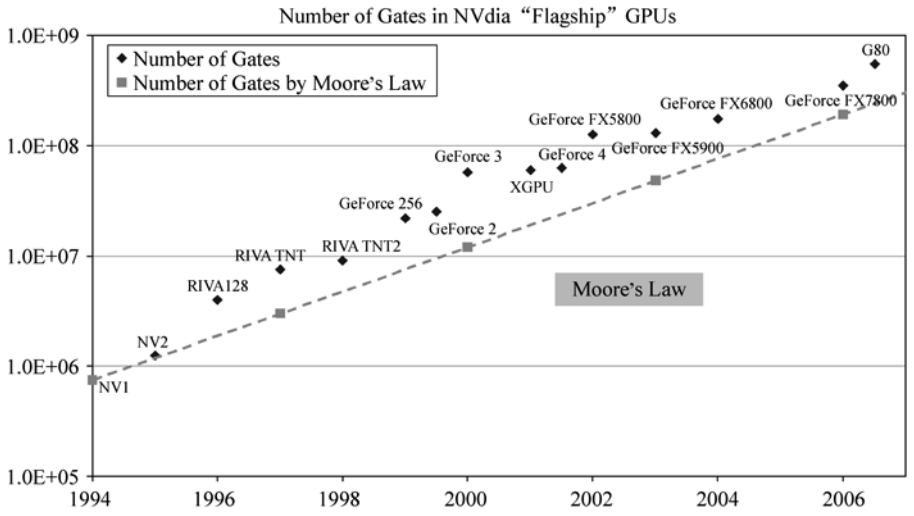


Figure 1.1 Actual chip complexity increases faster than Moore's law

Despite the strong need for more silicon real estate, the basic physics laws would not allow an unlimited scaling of device dimension. The limit would have to be

reached in the next 10 – 20 years, if no replacement technologies come up during this time frame.

Meanwhile, emergent very large scale integration (VLSI) systems are incurring overwhelming complexity as the main-stream process technology is now moving to the 45 nm node. Among many difficulties, the following three problems are inherent to the very nature of monolithic integration:

Interconnection Performance Historically, the functionality to be integrated in a single chip at every technology generation has always exceeded the capacity provided by pure scaling. To accommodate the extra transistors, the chip size has always been increasing since the invention of the first IC[3]. The problem is that, the interconnection length, especially worst-case interconnection length, has to increase accordingly. Starting from the 0.25 μm technology node, the interconnection delay of long on-chip wires has become the dominant part determining system performance[3]. Unfortunately, interconnection delay is very hard to predict before the circuit is actually laid out. As a result, IC architects usually take considerable efforts to manage those long wires with the help of advanced electronic design automation (EDA) software.

Mixed Technology Integration Modern System-on-Chips (SoCs) typically have to integrate heterogeneous, mixed-technology components. The technology heterogeneity certainly complicates the underlying fabrication processes. The fabrication cost of today's semiconductor processes is already skyrocketing with the shrinking of the feature size[4]. A single mask set as well as the corresponding probe for digital ASICs is reported to soon reach $5 million at the 45 nm technology node[5,6], while the price of a finished wafer in a RF-CMOS process is higher than that in a pure CMOS process by at least 15%[7]. Meanwhile, it is worth mentioning that certain RF circuits would not benefit from a finer process

in terms of performance improvement and cost reduction. For instance, some analog transistors and passive components (e.g., inductors) have to occupy a relatively constant die area to meet performance requirements no matter in which technology node they are fabricated[7].

Memory Wall Memory bandwidth has already become the limiting factor impeding the performance of general-purpose microprocessors and multimedia appliances, as well as other data-intensive applications. It has been reported that the processor performance has been improving by 35% annually from 1980 to 1986 and by 55% annually thereafter[8,9]. In the same period, the access latency of DRAM has been improving by only 7% per year[8]. The mainstream solution to this problem is to introduce cache hierarchy and/or integrate memories with the logic on the same chip. For most current processors, at least 50% of the die area is occupied by cache[10]. It is also estimated that a personal digital assistant (PDA)-type phone could use as much as 128 Mb flash and 128 Mb DRAM[11]. Embedded memories (especially embedded DRAM) require a merged memory/logic process, which is more expensive and leads to inferior memory devices[12,13]. Moreover, the long interconnects of the memory buses can also become a bottleneck when memory blocks become larger.

Summarizing the aforementioned concerns, the key question that arises is how to build modern VLSI systems that avoid the shortcomings of monolithic SoC, while maintaining momentum in the increase of the functionality? The work reported in this book tries to answer this question by considering a non-monolithic VLSI integration style, so-called the 2.5-D integration. In this first chapter, we will define the concept of the 3-D integration paradigm and then outline the organization of this book.

1.1 2.5-D Integration

The 2.5-D integration scheme is actually a revision of the concept, smart substrate multi-chip module (MCM), proposed in the past[14,15], but enhanced with the new feature of 3-D stacking of IC chips. To implement a VLSI system using the 2.5-D integration scheme, architects would partition the system into a number of subsystems, each containing components that are going to be fabricated in a specific technology. Logic and layout implementations for every sub-system are performed such that each cluster can be fabricated as an unpackaged die, optimized for performance, cost and/or power consumption. Finally these chips can be stacked together in the manner, for instance, as illustrated in Fig. 1.2. In this particular implementation the inter-die communication and power distribution might be accomplished through 'vertical' interconnects between stacked dies. In this book, we designate the vertical interconnect as 'inter-chip contact'.

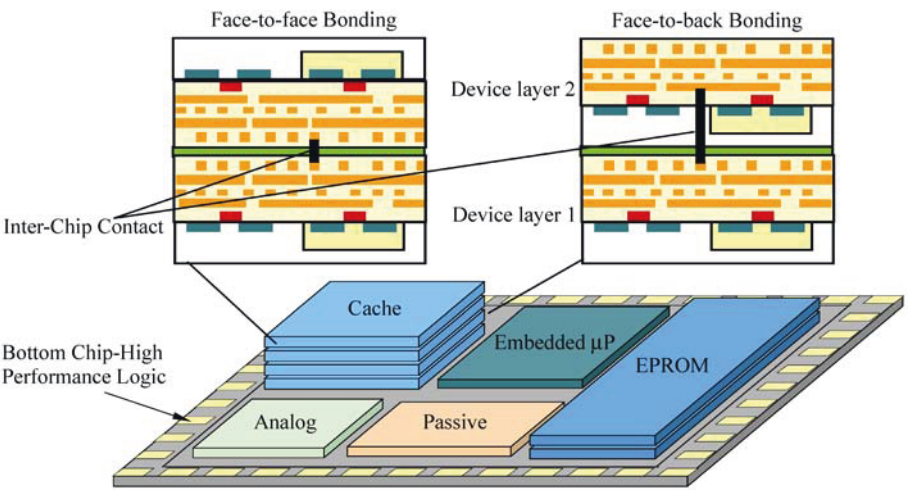


Figure 1.2 An imaginary 2.5-D system (*see colour plate*)

One of the major stumbling blocks of MCM is the 'Known Good Die' problem. It refers to the fact that the bare dies, which are required by MCM level integration, generally do not provide enough test coverage at a cost-efficient level. To overcome this 'imperfect testing' problem associated with the MCM approach, the 2.5-D integration scheme must be enhanced with an incremental, hierarchical testing and assembling methodology[15]. According to our testing methodology, the bottom-level chip can be designed and fabricated with relatively sufficient testing support (e.g., partially packaged), while the upper layer chips can be designed with adequate self-testing logic and isolation capability. Thus, the bottom level chip can be properly tested during the assembling process and then used as a chassis to test the upper layer chip(s). Next each die at the upper levels can be tested as soon as it is 'plugged' into a partially assembled 2.5-D system using available hardware/software components. In addition, if the stacking process can be designed to support rework to a given extent, it's possible to replace a faulty die with a new one without damaging the whole system.

Intuitively, many advantages can be expected through the adoption of 2.5-D integration scheme. A few of these are discussed below.

Smaller System Footprint By eliminating intermediate packaging levels, the 2.5-D integration scheme allows a system to be constructed with a much smaller volume. The removal of packages also helps reduce the system weight. Both advantages have an important implication for portable computing and communicating devices.

Reduced Interconnection Length Generally speaking, the underlying topology of VLSI circuits is not planar. Hence a planar embedding of a VLSI system into a monolithic surface will lead to overhead in the interconnection length even with many wiring layers. On the other hand, 2.5-D integration enables designers or

CAD tools to find a more efficient packing of circuits according to their inherent topology. Thus, a systematic reduction in the on-chip wire length can be expected, which can be translated into speed gain, power saving, and many other advantages.

Decoupling Between Functionality Increase and Technology Selection Traditionally, the CMOS process has been the only viable vehicle for true system integration because a dominant percentage of a system could be cost-efficiently manufactured only this way. Under the 2.5-D paradigm, a VLSI system can be properly divided into multiple chips so that the fabrication cost and system performance can be optimized. For instance, for a wireless application built using 2.5-D integration concept, the RF transceiver circuit can be manufactured with a SiGe-BiCMOS process with excellent RF performance, especially for sensitivity and low-power consumption. Meanwhile, high-performance digital signal processing circuits can be built with a high-speed CMOS process, while other logic circuits for user-applications can be built with a high VT, low-power CMOS process. Finally, high-density, low-power DRAMs fabricated with a dedicated DRAM process can be stacked on the top of the logic chips to store large volume of data.

New Opportunity for Reuse The 2.5-D integration scheme enables reusing verified components at a die level. For instance, IP cores can be delivered as pre fabricated and fully characterized dies with standard interfaces. A family of VLSI systems for a given application but with different performance/cost targets can be realized as different combinations of standard IP dies. From a design perspective, integrating these IP-dies only requires Lesser effort than the IP-core integration does. This new paradigm of IP reuse promises that VLSI systems could be designed and implemented in significantly reduced time with a considerably lower system cost.

1.2 Enabling Technologies

In the previous sections, we introduced the concept 2.5-D integration. We believe that the success of such a new VLSI integration scheme depends on the synergy of three key enabling technologies: fabrication technology, testing methodology, and design technology. In this section, we briefly review the status of these technologies and identify the problems that need to be resolved in the future.

1.2.1 Fabrication Technology

Under the 2.5-D integration context, a complete VLSI system is an assembly of fabricated, unpackaged dies. Although each die can be fabricated with a conventional technology, the central problem is how to vertically bond chips and construct inter-chip contacts in a yield-efficient way. One potential solution can be developed on the basis of the wafer bonding technology, which has attracted considerable research work (e.g., [17–40]). However, the wafer bonding technology poses very high accuracy requirements (better than 1 micron[41]) for the aligners. Such high precision aligners tend to have prohibitive equipment cost[41]. In addition, current high-precision aligners are being mainly developed for the MEMS industry and typically need double-side processing of wafers, which is not applicable under the wafer bonding context. Aligners designed for wafer bonding are being developed (e.g., [42,43]), but are still in R&D or early commercial stages.

Accordingly, we envision a bonding technology utilizing a passive, high-precision, self-alignment mechanical latch, e.g., a laterally compliant cantilever with a contact clamp. Borrowing techniques from the MEMS community, the alignment process can be organized into multiple stages with increasing accuracies. The refined

alignment stage can potentially have very high precision since the alignment features are fabricated in the same process step as the top metallization layer in CMOS technology.

1.2.2 Testing Methodology and Fault Tolerance Technique

To achieve a cost-efficient fabrication process, a 2.5-D system has to be assembled and tested in an incremental manner so that sufficient fault coverage level can be achieved with a reasonable cost. Such a methodology actually was developed when the idea of 2.5-D integration was first proposed[15,16]. Using this testing methodology, every die in the system is isolated from the remaining dies of the system by a dedicated boundary-scan chain and can be selectively powered. These features make it possible to separately and incrementally test every die in either a fully or a partially assembled system.

Meanwhile, validating a die in a 2.5-D system is quite similar to verifying an embedded IP core from the testing point of view. Thus, recently developed testing methods for core-based designs[44] provide another set of testing solutions for the 2.5-D system as listed below:

- Core isolation techniques such as partial boundary scan chain[45] and test wrapper techniques[46].
- Test data propagation to and from a specific core by set other cores into a transparent mode[47].
- Reuse of system resource like system bus[48] for test purpose.
- Utilization of in-system microprocessor to perform self-testing[49].
- Testing solution for embedded memories[50,51].

The essence of the above techniques is to enable separate access to each embedded IP core in a system while trying to reuse existing functionality as much as possible. Accordingly, these testing solutions can be straightforwardly adapted to the purpose of testing 2.5-D systems.

Another approach to overcome the difficulty of test access in a 2.5-D system is to use fault tolerance techniques. For a 2.5-D stacked system, redundant components can be extensively deployed at different granularity levels to compensate the difficulty of testing access. Historically, the fine-grained techniques, such as employing redundant rows or columns of cells in array-styled circuits, were very successful[52]. On the other hand, coarse-grained techniques (e.g., replicating chip-level functional blocks) have not become popular because global failure tends to impact all modules (including redundant modules) simultaneously[52]. For instance, a short between power and ground in a functional block will lead to the failure of the whole system no matter how many redundant blocks are installed. However, coarse-grained techniques may prove to be very useful in a 2.5-D system since functional blocks at different device layers can be fully decoupled.

1.2.3 Design Technology

With the maturation of 3-D stacking technologies, an essential issue is to develop corresponding CAD tools and design flows due to involvement of a large number of design variables (a partial list is given in Table 1.1). Besides traditional performance requirements and design constraints, a design automation framework for 2.5-D integrated systems has to take into account the following factors discussed below.

Table 1.1 Design variables involved in designing a 2.5-D system

DESIGN VARIABLES	OPTIONS
Number of device layers	1*, 2, 3, …
Process	Standard CMOS, RF-CMOS, SiGe, Embedded memory
Inter-chip contact density	Fixed in certain processes, but can also be tuned to balance performance and manufacturability in other technologies
Cell library	High performance, low power
Design technology	Digital, analog, optical, MEMS
IP reuse	Hard/Firm/Soft IP cores, die-Level IPs, Bus standard, Embedded operation system
Memory organization	Memory hierarchy, number of banks per block, cache associativity, memory/cache bus width/line size, memory interface protocol
Communication protocols	Bus, on-chip networks, asynchronous, global asynchronous local synchronous (GALS)
Heat dissipation feature	Heat sink, unconnected inter-chip contact, MEMS micro-pipe

* Monolithic SoC can be considered as a special case of 2.5-D system with only one device layer.

Inter-chip Contacts The inter-chip contacts constitute a new level of inter-connection hierarchy. It is crucial to efficiently utilize this physical resource of the stacked system in a systematic manner. In fact, if we assume the inter-chip contacts can be placed everywhere on the chip surface and have an area pitch of 5 μm by 5 μm, then up to 4 million of them will be available on a chip with a die area of 1 cm by 1 cm. This large amount of communication resource suggests that 2.5-D integration is no longer a packaging option but a design opportunity. In addition, the dimension of inter-chip contacts can be treated as a design variable. In other words, the dimension of inter-chips should be adjustable for different designs (but uniform for a given design). A larger dimension offers better reliability

and manufacturability. A smaller size leads to a higher density and thus potentially higher system performance. Accordingly, IC architects must achieve a balance of performance and yield by choosing a proper dimension for inter-chip contacts.

Thermal and Reliability Issues Modern monolithic VLSI designs already have to consider heat dissipation and reliability issues. In the 2.5-D system, these issues will be even exaggerated. One crucial problem is the heat dissipation in the upper layer chips that are not directly attached to a heat sink. Excessive heat may stress the system and lead to poor mean time to failure (MTTF). In addition, transistors working under a higher temperature also generate a larger leakage current. The electromagnetic noise will also be an essential concern for 2.5-D systems due to the reduction of isolation space between devices. For example, the reliability of DRAM and mixed-signal circuitry can be severely affected by the noise generated in the upper and/or lower layer chips.

Layout Synthesis Tools for 2.5-D Integration The 2.5-D floorplanning, placement and routing tools can be constructed by extending from existing algorithms. These 2.5-D-aware tools determine the geometric features of a designed system in a stacked space and establish how to assign inter-chip contacts into design hierarchy according to the internal structure of a designed system. An important issue is to consider thermal issues like hot-spot removal and heat dissipation feature placement.

Physical Verification Tools Current design rule checking (DRC) tools need to be enhanced to handle the design rules associated with inter-chip contacts. In addition, the parasitic extraction tools have to consider coupling between adjacent device layers as well as the RLC parameter of inter-chip contacts. One important concern is the electromagnetic noise in the whole system since isolation among different devices is hard to guarantee.

Automatic Design Exploration An automatic exploration engine is crucial to help designers conquer the complexity of 2.5-D system integration. Important design variables such as number of layers of chips, technology selection for each slice of chip, density of inter-chip contact, and system partitioning, should be determined through searching the complex solution space in an automatic manner. We envision the exploration engine could extract estimations of performance, manufacturability and power consumption from a physical prototype constructed by fast register-transfer level (RTL) synthesis and coarse-grained placement.

Memory Architecture Navigation Increasing percentage of silicon is likely to be devoted to memory in future VLSI systems. A 2.5-D integrated system could combine multiple types of memory blocks (e.g., DRAM, Flash, EEPROM, etc.) with different I/O protocols (SDRAM, DDR, RDRAM) and internal organizations. Different parts of memory blocks can be interconnected with a 3-dimensional network by taking advantage of inter-chip contacts. In addition, memory blocks should be organized in such a way that internal blocks are separately shut down to avoid unnecessary power consumption. As a result, memory architecture navigation algorithms are critical to find optimum memory configuration.

1.3 Objectives and Book Organization

As indicated in the previous sub-sections, IC industry will have to take major efforts to develop the 3-D stacking technologies. In this book we will focus on a small but important set of elements of these efforts that may be useful to initiate activities in design technology for 2.5-D integration. Specifically, the objective of this research is to provide a thorough feasibility study on the 2.5-D integration

strategy. We are going to approach the problem in three consecutive steps as listed below.

Cost Analysis First, we are going to establish a unified cost analysis framework to compare the fabrication cost of different integration schemes. In fact, given a VLSI application, there are many different ways to build it on silicon. Accordingly, the first research objective is to provide an apple-to-apple cost comparison for different integration styles.

2.5-D Design Case Studies Next we plan to use design cases studies to prove the potential of the 2.5-D integration strategy. We will first focus on custom designs to explore such key design attributes such as geometrical characteristics of a layout implementation, timing performance, and system level performance in terms of processor throughput. Another goal is to identify major design constraints introduced by the 2.5-D stacking technology.

2.5-D ASIC Physical Design Automation For ASIC designs, it's not feasible to find an optimized 2.5-D implementation manually. Therefore, we have to build the first-generation 2.5-D layout design tools to automatically implement stacked layout. Then we can compare the interconnect characteristics between the monolithic and 2.5-D layout implementations of a give system and thus assess the feasibility of the 2.5-D paradigm.

In the remaining parts of this book, we would cover our journey toward a feasibility evaluation on the 2.5-D integration paradigm. The succeeding chapters are organized as follows:

In Chapter 2, we present a unified cost analysis among five different integration schemes: monolithic System-on-Chip, Multiple-Reticle-Wafer, Multi-Chip Module, 2.5-D integration, and 3-D integration. Our results proved that the 2.5-D integration scheme could be the most cost efficient under a group of reasonable assumptions.

After assessing the cost tradeoffs, we evaluate the feasibility of the 2.5-D concept through a series of design case studies. These studies demonstrate how to explore a solution space including both monolithic and 2.5-D integration schemes. In the 2.5-D implementations of these systems, circuit/system architectures are determined according to the design attributes of target applications. These studies establish that the 2.5-D integration offers important opportunities for performance improvement.

Based on the experiences gathered from the design case studies, we extend the feasibility study to general designs in the following three chapters. To study the feasibility of the 2.5-D integration strategy for a design application, we compare the interconnection characteristics between its monolithic and 2.5-D layout implementations. We evaluate the interconnection characteristics at two different abstraction levels: floorplan and placement levels, for a complete view of the whole design trajectory. In modern VLSI designs, long wires would determine the timing performance. If we observe improved wire length distribution in the 2.5-D layout, it can be concluded that the 2.5-D implementation could outperform its monolithic equivalents.

Due to the complexity of modern VLSI systems, we had to develop 2.5-D layout synthesis tools to automatically pack a system in the 2.5-D layout space. In Chapter 4, we outline our 2.5-D physical design flow and basic assumptions. In addition, we introduce our global router that provides constructive wire length estimation during the design flow. Chapter 5 and 6 cover the feasibility study on the floorplan level and placement level, respectively. In these two chapters, the classic physical design problems have been re-formulated under the 2.5-D integration context. Then we discuss our prototyping 2.5-D physical design tools. A large number of design case studies are used to justify the feasibility of the

2.5-D paradigm.

Based on the above work, we propose a roadmap for the application of 2.5-D integration scheme in Chapter 7. Proposing a time line for the adoption of 2.5-D integration, we also discuss major design tradeoffs for several important categories of VLSI applications. We conclude the book in the final chapter by summing up our major contributions and outlining future research directions in fabrication, test, and design technologies.

Finally, in Chapter 8, we summarize our work on 2.5-D system integration. We also discuss future research directions to convert 2.5-D integration scheme into real-world industrial practices.

References

[1] NVidia Corporate, [online]. Available: http://www.nvidia.com/page/home.html.

[2] J. M. Rabaey. Wireless beyond the third generation—facing the energy challenge. In: Proc. Int'l Symposium of Low Power Electronic Devices, 2001, pp. 1 – 3.

[3] J. D. Plummer, M. D. Deal, P. B. Griffin. Silicon VLSI technology. Prentice Hall, New Jersey, 2000, Ch. 1.

[4] W. Maly. IC design in high-cost nanometer-technologies era. In: Proc. Design Automation Conf., 2001, pp. 9 – 14.

[5] M. Ooishi. TSMC Takes Lead in 45 nm IC Mass Production. Nikkei Electronics Asia, July 2007, [online]. Available: http://techon.nikkeibp.co.jp/article/HONSHI/20070626/134824/.

[6] M. LaPedus. Foundries face obstacle course in 45 nm race. EE Times Asia, May, 2007, http://www.eetasia.com/ART_8800464631_480100_NT_d25bb9ea.HTM.

[7] A. Matsuzawa. RF-SoC—expectations and required conditions. IEEE Trans. on Microwave Theory and Techniques, Vol. 50Jan. 2002, pp. 245 – 253.

[8] J. Hennessy, D. Patterson. Computer architecture: a quantitative approach. Third edition, Ch. 5 Morgan Kaufmann, San Francisco, CA, 2002.

[9] D. Burger, J. Goodman. Memory bandwidth limitations of future microprocessors. In: Proc. 23rd Int'l Symposium on Computer Architecture, May 1996, pp. 78 – 89.

[10] S. Rusu, et al.. A 1.5-GHz 130-nm Itanium 2 processor with 6-MB on-die L3 cache. IEEE Journal of Solid-State Circuits, Vol. 38, Nov. 2003, pp. 1887 – 1895.

[11] G. Purvis. Demands on 3G memory set to soar. Wirelessweb, [online]. Available: http://wireless.iop.org/articles/feature/2/6/2/1.

[12] D. Keitel-Schulz, N. When. Embedded DRAM development: Technology, physical design, and application issues. IEEE Design & Test of Computers, May 2001, pp. 7 – 15.

[13] N. When, S. Hein. Embedded DRAM architectural trade-offs. In: Proc. Design, Automation and Test Europe, 1998, pp. 704 – 708.

[14] W. Maly, et al.. Smart-substrate multichip-module systems. IEEE Design & Test of Computers, Vol. 11, Summer 1994, pp. 64 – 73.

[15] A. Gattiker, W. Maly. Smart substrate MCMs. Journal of Electronic Testing Theory and Applications (JETTA), Vol. 10, 1997, pp. 39 – 53.

[16] W. Maly. Prospects for WSI: a manufacturing perspective. IEEE Computer, Feb. 1992, pp. 39 – 53.

[17] S. Kawamura, N. Sasaki, T. Iwai, M. Nakano, M. Takagi. Three-dimensional CMOS ICs fabricated by using beam recrystallization. IEEE Electron Device Lett., Vol. EDL-4, Oct. 1983, pp. 366 – 368.

[18] M. Nakano. 3-D SOI/CMOS. In: Proc. Int'l Electronic Device Meeting, 1984, pp. 792 – 795.

[19] S. Kawamura, N. Sasaki, T. Iwai, M. Nakano, M. Takagi. Three-dimensional CMOS ICs fabricated by using beam recrystallization. IEEE Electron Device Lett., Vol. EDL-4, Oct. 1983, pp. 366 – 368.

[20] K. Banerjee, S. J. Souri, P. Kapur, K. C. Saraswat. 3-D ICs: a novel chip design for

improving deep-submicrometer interconnect performance and systems-on-chip integration. Proceedings of the IEEE, Vol. 89, 2001, pp. 602 – 633.

[21] V. Subramanian, M. Toita, N. R. Ibrahim, S. J. Souri, K. C. Saraswat. Low-leakage Germanium-seeded laterally-crystallized single-grain 100 nm TFTs for vertical integration applications. IEEE Electron Device Lett., Vol. 20, Jul. 1999, pp. 341 – 343.

[22] G. W. Neudeck, S. Pae, J. P. Denton, T. Su. Multiple layers of silicon-on-insulator for nanostructure devices. J. Vac. Sci. Technol. B, Vol. 17, No. 3, 1999, pp. 994 – 998.

[23] A. Heya, A. Masuda, H. Matsumura. Low-temperature crystallization of morphous silicon using atomic hydrogen generated by catalytic reaction on heated tungsten. Appl. Phys. Lett., Vol. 74, No.15, 1999, pp. 2143 – 2145.

[24] T. H. Lee. A vertical leap for microchips. Scientific American, Jan. 2002. [Online]. Available: http: //www.sciam.com/article.cfm?articleID = 000BD05C-D352-1C6A-84A9809EC588EF21&sc = I100322.

[25] M. Chan. The potential and realization of multi-layers three-dimensional integrated circuit. Int'l Solid-State and Integrated-Circuit Technology, 2001, 40 – 45.

[26] S. A. Kuhn, M. B. Kleiner, P. Ramm, W. Weber. Performance modeling of the interconnect structure of a three-dimensional integrated RISC processor/cache system. IEEE Trans. On Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, Vol. 19, Nov. 1996, pp. 719 – 727.

[27] S. A. Kuhn, M. B. Kleiner, P. Ramm, W. Weber. Performance improvement of the memory hierarchy of RISC-systems by application of 3-D technology. IEEE Trans. On Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, Vol. 19 No. 4, Nov. 1996, pp. 709 – 718.

[28] K. W. Lee, T. Nakamura, T. Ono, Y. Yamada, T. Mizukusa, H. Hashmoto, K. T. Park, H. Kurino, M. Koyanagi. Three-dimensional shared memory fabricated using wafer stacking technology. In: Proc. Int'l Electronic Device Meeting, 2000, pp. 165 – 168.

[29] M. Koyanagi, H. Kurino, Kang Wook Lee, K. Sakuma, N. Miyakawa, H. Itani. Future system-on-silicon LSI chips. IEEE Micro, Vol. 18, July-Aug. 1998, pp. 17 – 22.

[30] S. Strickland, E. Ergin, D. R. Kaeli, P. Zavracky. VLSI design in the 3rd dimension. Integration: the VLSI Journal, Vol. 25/1, Sep. 1998, pp. 1 – 16.

[31] S. M. Alam, D. E. Troxel, C. V. Thompson. A comprehensive layout methodology and layout-specific circuit analyses for three-dimensional integrated circuits. In: Proc. Int'l Sym'm on Quality Electronic Design, 2002, pp. 246 – 251.

[32] J. -Q Lu, et al.. A wafer-scale 3-D IC technology platform using dielectric bonding glues and copper damascene patterned inter-wafer interconnects. In: Proc. Int'l Interconnect Technology Conf., 2002, pp. 78 – 80.

[33] H. B. Pogge. The next chip challenge: effective methods for viable mixed technology SoCs. In: Proc. Design Automation Conf., 2002, pp. 84 – 87.

[34] K. W. Guarini, et al.. Electrical integrity of state-of-the-art 0.13 mm SOI CMOS devices and circuits transferred for three-dimensional (3-D) integrated circuit (IC) fabrication. In: Proc. Int'l Electronic Device Meeting, 2002, pp. 943 – 945.

[35] M. Ieong, et al.. Three dimensional CMOS devices and integrated circuits. In: Proc. Custom Integrated Circuits Conf., 2003, pp/ 207 – 213.

[36] B. Burari. Bridging the gap between the digital and real worlds: the expanding role of analog interface technologies. In: Proc. Solid-State Circuits Conference, 2003, pp. 30 – 35.

[37] J. Mayega. 3-D direct vertical interconnect microprocessors test vehicle. In: Proc. Great Lakes Symposium on VLSI, 2003, pp. 141 – 146.

[38] T. Mimura et al.. System module: a new chip-to-chip module technology. In: Proc. Custom Integrated Circuits Conf., 1997, pp. 439 – 442.

[39] Matrix Semiconductor. 3-D technology. [online]. Available: http://www.matrixsemi.com/index.shtml.

[40] Ziptronix Inc.. 3-D integration. [online]. Available: http://www.ziptronix.com/.

[41] P. Lindner, V. Dragoi, T. Glinsner, C. Schaefer, R. Islam. 3-D Interconnect through aligned wafer level bonding. In: Proc. IEEE Electronic Components and Technology Conf., 2002, pp. 1439 – 1443.

[42] R. Islam, C. Rrubaker, P. Lindner, C. Schaefer. Wafer level packaging and 3-D interconnect for IC technology. In: Proc. IEEE/SEMI Advanced Semiconductor Manufacturing Conf., 2002, pp. 212 – 217.

[43] T. Suga, M. M. R. Howlader, T. Itoh. A new wafer-bonder of ultra-high prision using surface activated bonding (SAB) concept. In: Proc. IEEE Electronic Components and Technology Con., 2001, pp. 1013 – 1018.

[44] Y. Zorian, E. J. Marinissen, S. Dey. Testing embedded-core based system chips. IEEE Computer, Jun. 1999, pp. 52 – 60.

[45] N. Touba, B. Pouya. Using partial isolation rings to test core-based designs. IEEE Design & Test of Computers, Dec. 1997, pp. 52 – 59.

[46] S. Koranne. Design of reconfigurable access wrappers for embedded core based SOC test. In: Proc. Int'l Symposium on Quality Electronic Design, 2002, pp. 106 – 111.

[47] I. Ghosh, N. Jha, S. Dey. A low overhead design for testability and test generation techniques for core-based systems. In: Proc. Int'l Testing Conf., No. 1997, pp. 50 – 59.

[48] M. Nicolaidis. IP for embedded robustness. In: Proc. Design, Automation and Test Europe, 2002, pp. 240 – 241.

[49] R. Rajsuman. Testing a system-on-a-chip with embedded microprocessor. In: Proc. Int'l Testing Conf., 1999, pp. 499 – 508.

[50] R. McConnell, R. Rajsuman. Test and repair of large embedded DRAMs I. In: Proc. Int'l Testing Conf., 2001, pp. 163 – 172.

[51] R. McConnell, R. Rajsuman. Test and repair of large embedded DRAMs 2. In: Proc. Int'l Testing Conf., 2001, pp. 173 – 181.

[52] W. Maly. Computer-Aided Design for VLSI Circuit Manufacturability. Proceedings of IEEE, Vol. 78, Feb. 1990, pp. 356 – 390.

2 A Cost Comparison of VLSI Integration Schemes

Yangdong (Steve) Deng

Institute of Microelectronics, Tsinghua University
Beijing 100084, P. R. China, dengyd@tsinghua.edu.cn

Wojciech P. Maly

Electrical and Computer Engineering Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891 USA, maly@ece.cmu.edu

Abstract Besides the dominant monolithic VLSI integration paradigm, many non-monolithic schemes have already been developed in the past. Typical such schemes include wafer scale integration or multi-reticle wafer, multi-chip module, and 3-D integration. In this chapter we compared these different schemes in a unified cost analysis framework. Our model takes a few parameters extracted from representative fabrication and evaluates the cost efficiency. Our analysis proves that the proposed 2.5-D out significantly outperform other integration paradigms from a cost perspective.

Keywords 2.5-D integration, monolithic VLSI integration, multi-reticle wafer, multi-chip module, 3-D integration, yield, silicon area, fault coverage.

Besides the 2.5-D integration paradigm introduced in the previous chapter, there have been many non-monolithic integration solutions proposed in the past to address the problems inherent to the monolithic integration scheme. Such non-monolithic VLSI integration schemes include Multiple-Reticle Wafer derived from the Wafer Scale Integration[1], Multi-Chip Module (MCM)[2], and 3-D integration[3]. Together with monolithic System-on-Chip, these integration schemes provide interesting tradeoffs to implement a given design application. Accordingly, it is essential to investigate the cost implications associated with these schemes and identify the most cost efficient one. In this chapter, we will establish a cost analysis framework to compare the cost tradeoff of different integration paradigms.

In the past, a large body work on the IC yield has been published and one excellent review is presented in [4]. The cost implications of Wafer Scale Integration and MCM have been extensively investigated in [5] and [2], respectively. Recently, Peng and Manohar proposed a simple binomial yield model for 3-D integrated ICs[6]. All the above research, however, focuses on the yield aspect of one specific integration style. To our best knowledge, the work reported in this chapter is the first one that could provide cost comparison for all the above mentioned integration schemes under a unified framework.

In this chapter, we first briefly review the concepts of major non-monolithic integration approaches. Then we describe an analytical cost analysis framework and then apply it to compare all 5 integration styles discussed before.

2.1 Non-Monolithic Integration Schemes

Before we can perform cost analysis, we need to introduce the concept of representative non-monolithic integration schemes. Besides the 2.5-D paradigm, here

we consider multiple-reticle wafer, multi-chip module, and 3-D integration.

2.1.1 Multiple-Reticle Wafer

The Multiple-Reticle Wafer approach is a revised version of wafer scale integration[1]. Under such a context, a system is partitioned into multiple dies and these dies are built on the same wafer. Since faulty die(s) may be generated during the manufacturing process, redundant dies have to be introduced to guarantee correct functioning of the system. However, one issue hindering this approach is the complexity introduced by multiple sets of masks when a VLSI application doesn't possess a regular architecture. Meanwhile, the redundant dies and corresponding voting logic and wires would lead to performance overhead. Thus this multiple-die scheme could hardly outperform its monolithic equivalent from a performance point of view.

2.1.2 Multiple Chip Module (MCM)

MCM system assembly[2] was extensively considered as an alternative packaging solution for VLSI systems. Under this context, bare dies are mounted on a common substrate, which can be simply a miniaturized PCB (MCM-L), a piece of glass ceramic (MCM-C), or alternating deposited layers of high-density thin-film metal and low dielectric materials (MCM-D). Commonly used techniques to bond the chips and the substrate include wire bonding, tape automated bonding (TAB), and flip-chip or controlled-collapse chip connections (C4). The problem here is that wafer probing testing is very difficult for at-speed tests and thus bare dies usually can only be partially tested. Due to the imperfect testing, faulty dies can

be introduced into a MCM system and lead to poor yield of the whole module. The standard way to address the problem is through the so-called “known good dies (KGDs)” which have a very high probability to correctly function. However, the techniques to guarantee the high fault coverage tend to be very expensive. On the other hand, MCMs could not be superior to monolithic ICs in terms of performance because long inter-die wires would still appear on the substrate. Of course, the substrate interconnects would be shorter than their board-level equivalents, and thus MCM is mainly used as a replacement for circuit boards in high-performance systems.

2.1.3 Three-Dimensional (3-D) integration

3-D ICs can be dated back to as early as 1980s (e.g., [7,8]) and has been followed by many recent developments[9]. Fabrication technologies for the 3-D integration can be classified into two categories: silicon re-growth and wafer bonding.

In the silicon re-growth approach, a new slice of silicon is formed on top of an existing substrate by either the following three methods: 1) polysilicon deposition (which can be used to build thin-film transistor)[10]; 2) epitaxy growth[11]; and 3) amorphous silicon deposition and crystallization[8,12,13]. The inter-chip contacts can be constructed by extending via-formation techniques, e.g., etching through the new layer of silicon. An important concern of this approach is that the formation of each new slice of silicon must be compatible with metal interconnects underneath. For Cu wires, the processing temperature must be under 450℃ so that the Cu diffusion effect doesn’t occur. Repeated exposure to high temperature also tends to impair the quality of transistors in the lower layer wafers.

Under the context of wafer bonding (e.g., [14−28]), wafers can be built with

traditional processes. Then an upper layer wafer is first grinded from back to a thickness of around 10 microns and bonded on the top of the lower layer wafer. To construct the inter-chip interconnection, one solution is to etch through-wafer vias all the way from the top level wafer to the uppermost metal layer on the lower layer of wafer. An alternative solution involves three steps: initially building the inter-chip interconnects as embedded contacts, then exposing them by grind the wafer from the back to a proper thickness, and finally bonding them with the bump built on the top of the lower layer wafer. Alignment accuracy determines the lower limit of the size of inter-chip interconnects. In the recently reported processes listed in Table 2.1, alignment accuracy is within the range of ±3 μm. Thus, the footprint of inter-chip contact could be as small as ~10 μm^2.

Table 2.1 Wafer bonding based 3-D integration technologies

Research Group	Stacking Style	Bonding Interface[1]	Inter-chip Contact		
			Alignment Accuracy (μm)	Footprint (μm^2)	Height (μm)
IBM[22–24]	Face-to-back	Adhesive	N/A	Variable	25 – 60
Tohoku Univ.[17,18]	Face-to-face or face-to-back	Adhesive	±1	3 × 3	~30
NEU[19]	Face-to-face	Adhesive	~±3	~10	<5
MIT[20]	Face-to-back	Cu-Cu interface	±3	(3~5) × (3~5)	<10
RPI/UAlbany[21]	Face-to-face or face-to-back	Low-k dielectric glue	±1~2	3 × 3	~30

[1] By bonding interface we indicate the manner in which two chips are attached.

Since it allows wafers built with different processes to be assembled, the wafer bonding technology is more flexible than the silicon growth approach. In addition, the stacking technology does not involve high-temperature processing. On the other hand, the wafer thinning step still could lower the fabrication yield or even lead to the failure of wafer manufacture.

2.2 Yield Analysis of Different VLSI Integration Approaches

As discussed in the previous section, there exist multiple integration styles to build a given VLSI system into silicon. In this section, we present a unified cost analysis framework to compare various integration strategies.

For a given semiconductor technology, the fabrication cost of a VLSI system can be measured by its total consumed silicon area, S_A, which is calculated as:

$$S_A = A / Y \tag{2.1}$$

where A is the actual silicon area, or the working silicon area, used by the VLSI implementation and Y is the fabrication yield.

The yield, Y, is given by the product of yield over all system components and layout layers[4]:

$$Y = \prod_{i=1}^{m} Y_i = \prod_{i=1}^{m} \prod_{j=1}^{n} Y_{ij} \tag{2.2}$$

where m is the number of components or dies in the system, n is the number of layers in the layout structure, Y_i is the yield of the i-th component, and Y_{ij} is the yield of the i-th component of the system due to the defects in the j-th layer of the IC layout structure. There exist a number of approaches to compute Y_{ij}[5]. In

this book we use a simple one[4]:

$$Y_{ij} = \mathrm{e}^{-A_{ij} \cdot D} \tag{2.3}$$

where A_{ij} is the area of i-th component on the j-th layer and D is defect density. It is generally assumed that layout structures on different layers have the same area. In other words, A_{ij} can be replaced as A_i (for $1 \leqslant j \leqslant n$) and we have:

$$Y_i = \mathrm{e}^{-n \cdot A_i \cdot D} \tag{2.4}$$

As a result, Equation (2.2) can be simplified as:

$$Y = \prod_{i=1}^{m} \mathrm{e}^{-n \cdot A_i \cdot D} \tag{2.5}$$

With the above yield model, we consider implementing a common application using different integration schemes. It is assumed that this application needs a die size of 4 cm^2 when manufactured as a monolithic chip1. For the integration schemes, the underlying manufacturing process is a 0.13 μm, 6-metal CMOS process with a wafer diameter of 300 mm. It is assumed that the wafer has a fabrication cost of \$2500[29]. In this process there are 19 layout layers: n-well, active region, n-select, p-select, thin oxide, polysilicon, oxide, 6 metal layers, and 6 inter-metal isolation layers. The defect density, D, is assumed to have a value of 0.025 particles/cm^2. The values of major parameters are shown in Table 2.2. For the non-monolithic integration approaches, we assume the design application could be partitioned into multiple parts with identical chip areas and all the parts could be manufactured with the same CMOS process. In the rest of this section we will discuss the cost implications of different VLSI integration strategies under the framework given by Equations (2.1) to (2.5).

Table 2.2 Values for the major parameters of our cost model

Symbol	Parameter	Value
D	Defect density	0.025 particles/cm^2
A	Silicon area when implemented as a single chip	4 cm^2
A_i	Silicon area of one component (die)	A/m
n	# Layers of layout structure	19
S_w	Wafer area	706.86 cm^2
C_w	Wafer cost	\$2500
C_t	Testing cost per second	\$0.12/s
k	Steepness of fault coverage with regard to time	0.116495
$C_{Carrier}$	KGD testing carrier cost	\$2.4
$F_{c\text{-}MCM}$	Fault coverage level of MCM	0.999
Y_a	3-D assembling yield	0.95
Y_R	Multi-reticle reconfiguration yield	0.99
O_M	Multi-reticle cost overhead	0.125 per extra die

2.2.1 Monolithic Soc

Given the parameter with values given in Table 2.2, the single-chip implementation ($m = 1$) of the system has a yield of only 15.0%, making it very cost-inefficient. According to Equation (2.1), the silicon area of the monolithic chip is 26.744 cm^2.

2.2.2 Multiple-Reticle Wafer (MRW)

In this approach, a VLSI system is partitioned into m parts and then each part is

fabricated as a separated die on the same wafer. For this discussion, the m parts are assumed to have identical die areas. To improve defect tolerance, r identical dies are built for every component. After one wafer is fabricated, faulty dies will be bypassed by reconfiguring the wafer-level interconnections. Here we simply assume the whole system will correctly function as long as at least one die of every component is fault free. As a result, the yield is given by the formula:

$$Y_{\mathrm{MRW}} = \prod_{i=1}^{m}[1-(1-Y_i)^r]\cdot Y_{\mathrm{R}} = \prod_{i=1}^{m}\left[1-\left(1-\mathrm{e}^{-\frac{A}{m}\cdot D}\right)^r\right]\cdot Y_{\mathrm{R}} = \left[1-\left(1-\mathrm{e}^{-\frac{A}{m}\cdot D}\right)^r\right]^m \cdot Y_{\mathrm{R}} \tag{2.6}$$

where Y_{R} represents the yield of the reconfiguration process. It bears mentioning that the fabrication time of a MRW (multiple-reticle wafer) is considerably longer due to the usage of multiple reticles[1]. We assumed that every die introduces a modest cost overhead of 12.5%, which is actually lower than the typical values for most real design cases[30]. This way the silicon area of the MRW approach is given by the formula:

$$S_{A-\mathrm{MRW}} = \frac{A\cdot(1+O_{\mathrm{M}}\cdot m)\cdot r}{Y_{\mathrm{MRW}}} \tag{2.7}$$

where O_{M} is the Multi-reticle cost overhead.

Using Equations (2.6) and (2.7), the total consumed silicon areas under different values of r are shown in Fig. 2.1. Compared with the monolithic integration approach, the redundancy can be quite effective to reduce system cost. Regarding the number of redundant copies, the most minimum is realized by installing one extra copy for each die.

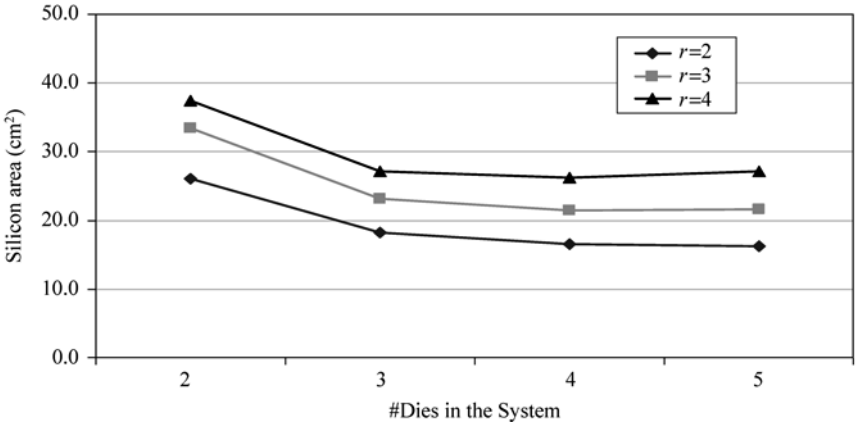


Figure 2.1 Total consumed silicon area of multiple-reticle wafer

2.2.3 Three-Dimensional (3-D) Integration

Within the paradigm of 3-D integration, the input VLSI system is built into *m* device layers, each having an equal area of *A/m*. The yield of the 3-D implementation is the accumulative yield over all layers:

$$Y_{3D} = Y_i \cdot \prod_{i=1}^{m-1} (Y_i \cdot Y_a) = Y_i^m \cdot Y_a^{m-1} \tag{2.8}$$

where Y_a is the yield loss due to the final 3-D assembling process. The factor Y^{m-1} is to take into account the fact that integration of *m* layers of chips requires (m – 1) silicon growth or wafer bonding procedures. This way the silicon area of the 3-D integration is thus given by:

$$S_{A-3D} = \frac{A}{Y_{3D}} = \frac{A}{Y_i^m \cdot Y_a^{m-1}} \tag{2.9}$$

The shortcoming of 3-D integration is obviously reflected by Equation (2.9).

While the yield loss has to accumulate in the fabrication process, the Y_a^{m-1} factor in the denominator suggests that the 3-D integration scheme is inherently more costly than the monolithic scheme. As a matter of fact, when the 3-D bonding step has an assembling yield of 95%, the total consumed silicon area of the 3-D implementation is 28.1 cm^2.

2.2.4 2.5-D System Integration

Under the 2.5-D integration context, a VLSI system is partitioned into *m* parts and then each part is fabricated as a separated die on different wafers. Finally these dies are assembled on a common substrate. Again we assume that every die has the same area, *A/m*.

As a result, the accumulative yield of one single die, $Y_{i\text{-2.5-D}}$, can be computed as the product of three components: (1) Y_i, which is the yield loss due to its own fabrication process; (2) Y_{Others}, the yield loss due to the assembling of other dies; and (3) Y_a, yield loss due to the final 3-D stacking process.

$$Y_{i-2.5\text{D}} = Y_i \cdot Y_{\text{Others}} \cdot Y_a \tag{2.10}$$

Y_i can be straightforwardly determined by Equation (2.5) and Y_a can be presupposed to have a constant value of 0.95. The computation of Y_{Others} depends on the fault coverage level of the dies (designated as F_C) in a 2.5-D system[5]:

$$Y_{\text{Others}} = (Y_i^{1-F_C})^{m-1} \tag{2.11}$$

where the purpose of the exponent $(m-1)$ is to take into account the yield loss due to the assembling of all other components.

On the other hand, the high fault coverage level comes with a price: extra testing time implies extra cost. Generally, a modest fault coverage level, e.g., 80%, can be achieved in relatively short testing time. However, a higher fault coverage level requires significantly increased testing time. Based on this observation, we propose to use an exponential model to correlate the test coverage level with the testing time, t:

$$F_{\mathrm{C}} = 1 - \mathrm{e}^{-k \cdot t} \tag{2.12}$$

where k is a constant defining the steepness of exponential function, which can be derived by assuming 60 seconds is long enough to achieve 99.9% fault coverage and 10% of total time is enough for 80% fault coverage. The testing cost, C_{Test}, can be assumed to be linearly proportional to the testing time:

$$C_{\mathrm{Test}} = C_{\mathrm{t}} \cdot t \tag{2.13}$$

In this research, we use a reasonable value of \$0.12 for C_{t}[31]. Meanwhile, since we already know the wafer cost and wafer size, the testing cost can be translated into silicon area by the formula:

$$S_{A-\mathrm{Test}} = \frac{C_{\mathrm{Test}}}{C_{\mathrm{w}}} \cdot S_{\mathrm{w}} = \frac{C_{\mathrm{t}} \cdot t}{C_{\mathrm{w}}} \cdot S_{\mathrm{w}} \tag{2.14}$$

From the above analysis, it can be seen that the silicon area of each die in a 2.5-D system consists of two components: fabrication silicon area and equivalent testing silicon area:

$$S_{Ai-2.5\mathrm{D}} = S_{Ai0} + S_{A-\mathrm{Test}} = \frac{A/m}{Y_{i-2.5\mathrm{D}}} + S_{A-\mathrm{Test}} \tag{2.15}$$

The silicon area of a 2.5-D system is the summation over all its components:

$$S_{A-2.5\mathrm{D}} = \sum_{i=1}^{m} S_{Ai-2.5\mathrm{D}} = \frac{A}{Y_{i-2.5\mathrm{D}}} + m \cdot S_{A-\mathrm{Test}} \tag{2.16}$$

The drawing in Fig. 2.2 shows the trend of silicon areas when the input VLSI application is partitioned into four layers and implemented as 4 separate dies. Clearly, fabrication cost decrease very rapidly with the increasing of fault coverage level achieved by the system components. However, it is shown that a fault coverage level that is very close to 100% is not necessary due to the associated excessive testing cost. For the application under discussion, the cost minimum is achieved at a 90% fault coverage level. Figure 2.3 illustrates silicon area values of the 2.5-D integration with different numbers of components/dies.

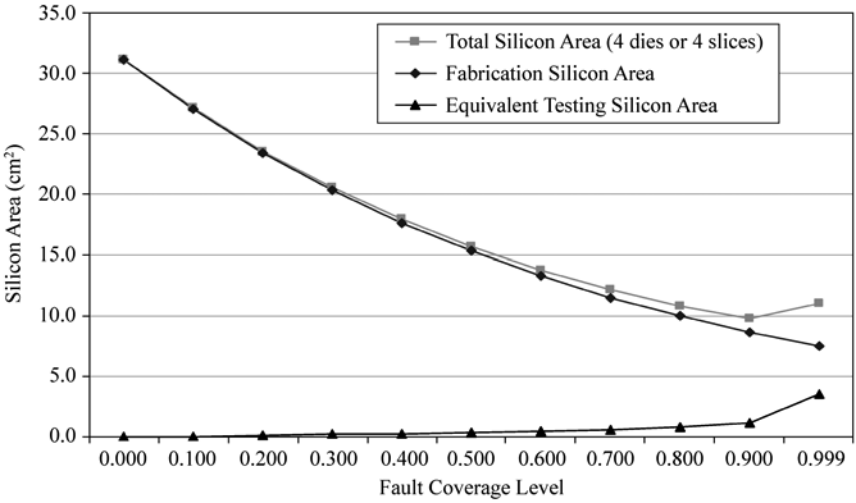


Figure 2.2 Silicon area of the 2.5-D implementation with 4 slices of chips

The above results lead to two important observations: (1) For dies with very incomplete or no testing at all, it's more cost efficient to implement the system in

smaller number of dies; and (2) For dies with reasonable (>40%) fault coverage level, 4 or 5 layers of dies, each with a die area of around 1 cm^2, would be the optimum choice from a cost perspective.

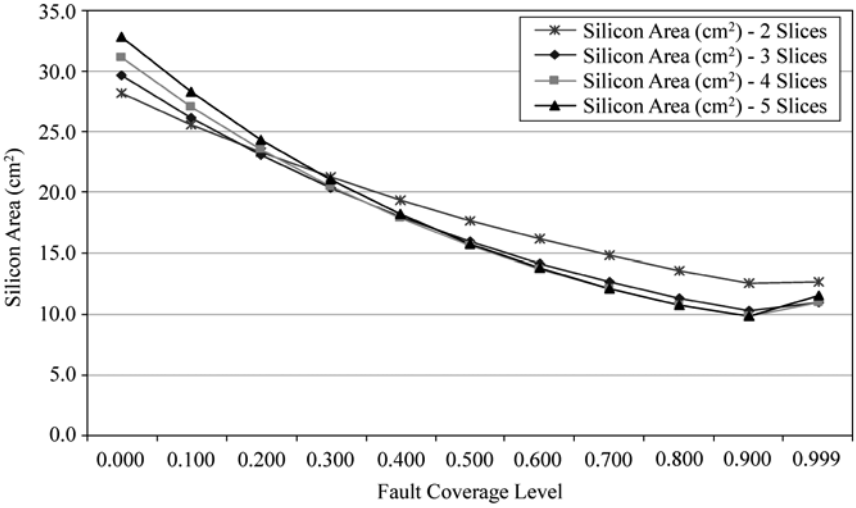


Figure 2.3 Silicon area of the 2.5-D implementation

2.2.5 Multi-Chip Module

From the silicon area perspective, the MCM scheme is similar to the 2.5-D scheme except the following 2 factors.

Very high fault coverage level Typically, MCM assembling usually requires know good dies with a fixed fault coverage level of 99.9% or even higher to ensure the correct functioning of the whole system. Such a high coverage level is necessary because the limited test access to internal components. Accordingly, we assume a fixed $F_C = 0.999$ in our computation to $S_{A\text{-Test}}$.

High test cost due to die carriers Due to the difficulty of test access,

MCMs often needs very expensive die carriers to achieve high fault coverage. In this work, the carrier cost as well as the extra test preparation time is modeled as having a cost equal to a testing time of 60 seconds[32]. As a result, the silicon area of MCM can be computed as:

$$S_{A-\mathrm{MCM}} = S_{Ai0} + S_{A-\mathrm{Test}} = \frac{A/m}{Y_{i-\mathrm{MCM}}} + S_{A-\mathrm{Test}} + S_{A-\mathrm{Carrier}} \tag{2.17}$$

where $Y_{i-\mathrm{MCM}}$ can be derived using Equation (2.10).

Based on the above analysis, the silicon areas of the MCM implementation for different numbers of dies can be computed and shown in Fig. 2.4.

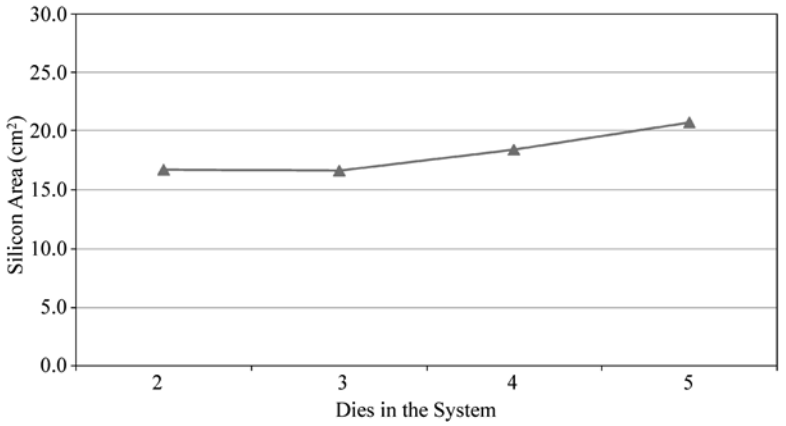


Figure 2.4 Silicon area of the MCM implementation

2.2.6 Summing Up

Finally, we pick up the cost optimum of each VLSI integration schemes discussed above and compare them in Fig. 2.5. To make a clearer comparison, we normalize the costs of different integration styles to that of the monolithic System-on-chip.

In Fig. 2.5, the Vertical-axis is the normalized silicon area consumed to fabricate a working system, while the Horizontal-axis is the defect coverage level of each system component or device level.

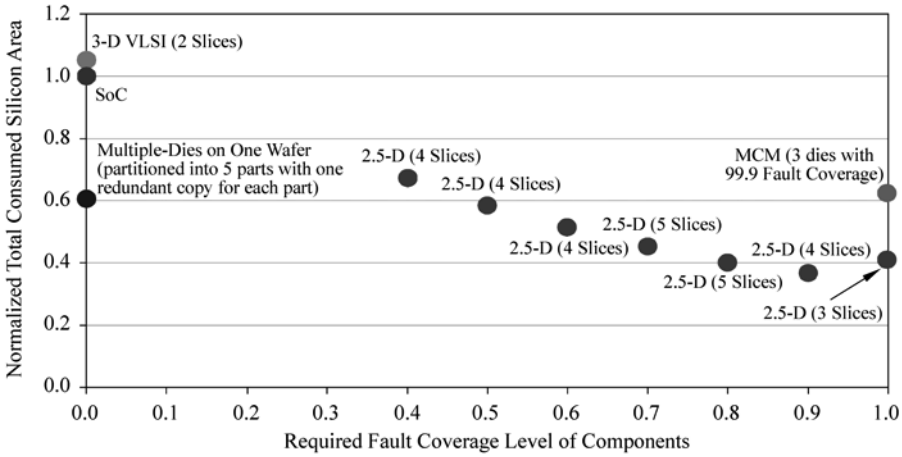


Figure 2.5 Silicon area comparison of different integration schemes

With the above set up, obviously, the system-on-chip built as a single chip will have a relative cost of 1.0 as well as a defect coverage level of 0 because no test is performed until the packaging stage. The multi-reticle wafer offers a smaller cost (i.e. ~60% of SoC) due to the help of redundant dies for each system component. The MCM approach has a similar cost, which is the result of using smaller, tested dies. For the 2.5-D scheme, we have the freedom of choosing different defect coverage levels. And the cost optimum is achieved when we partition the system into 4 parts and test each part with a defect coverage level of 90%. The cost reduction comes from two sources: (1) the ability of assembling and dissembling, and (2) system partition into smaller, tested dies. Finally, we can see the 3-D integration is the most expensive because the yield loss could only accumulate in the extra process steps to form the 3-D integration structures.

2.3 Observations

According to the cost analysis presented in this chapter, one can obviously see that the cost advantage of 2.5-D integration over other integration schemes. The cost potential is realized when proper test coverage can be achieved. In fact, given our assumptions, the 2.5-D implementation could be less expensive than the monolithic implementation by a factor of 2. This is an important finding which was not so obvious in the past. Such a result could be "discovered" for the first time because we have included in our analysis yield, cost of test, and test coverage as components of overall cost function.

On the other hand, the cost analysis framework can still be improved in many different aspects. For instance, now it is assumed that a system would be partitioned into multiple parts with an identical chip area and all the parts would be fabricated within the same CMOS technology. In the future, we should be able to analyze the case where a system is cut into multiple parts with varying sizes. Meanwhile, the cost analysis engine should be able to consider heterogeneous integration by taking into account the wafer costs for different fabrication processes (e.g., CMOS logic, memories, RF CMOS, GaAs, etc.). In addition, a more detailed cost analysis framework should also consider the fine-grained cost for individual processing steps so that the cost implication of a merged process could be assessed.

Our cost analysis framework provides a starting point for future system-level exploration (e.g., technology advisor) tools. In the future, with the maturation of 2.5-D and 3-D integration technologies, system architects will need to choose an integration paradigm as well as the corresponding system partition solution for a given design application. They will have to depend on automatic exploration tools to simultaneously considering multiple performance objectives including timing, cost, power, and so on.

A simplified flow for such a system exploration environment is shown in Fig. 2.6. For a given system configuration, a floorplanning tool could be used to extract related physical information. Then different feasibility analysis tools could be applied to evaluate the performance and cost of a given integration solution. Our cost analysis framework could be integrated to answer the what-if questions about the fabrication cost so that a cost efficient implementation could be identified. Of course, many other analysis engines are also required and we will further discuss these issues in the last chapter.

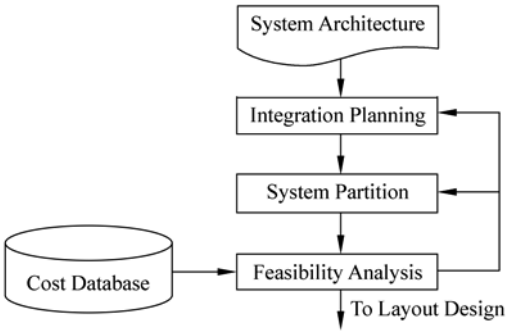


Figure 2.6 System planning for future VLSI systems

References

[1] W. Maly. Prospects for WSI: a manufacturing perspective. IEEE Computer, Feb. 1992, pp. 39 – 53.

[2] N. A. Sherwani, Q. Yu, S. Badida. Introduction to multi-chip modules. Wiley-Interscience, Nov. 23, 1995, Ch. 1.

[3] S. Kawamura, N. Sasaki, T. Iwai, M. Nakano, M. Takagi. Three-dimensional CMOS ICs fabricated by using beam recrystallization. IEEE Electron Device Lett., Vol. EDL-4, Oct. 1983, pp. 366 – 368.

[4] W. Maly. Computer-Aided Design for VLSI Circuit Manufacturability. Proceedings of IEEE, Vol. 78, Feb. 1990, pp. 356 – 390.

[5] W. Maly. Feasibility of Large Area Integrated Circuits. In: Wafer Scale Integration, edited by E.E. Swartzlander, Jr., published by Kluwer Academic Publishers, Boston, 1989.

[6] S. Peng, R. Manohar. Yield enhancement of asynchronous logic circuits through 3-dimensional integration technology. In: Proc. 16th ACM Great Lakes Symposium on VLSI, 2006, pp. 159 – 164.

[7] M. Nakano. 3-D SOI/CMOS. In: Proc. Int'l Electronic Device Meeting, 1984, pp. 792 – 795.

[8] S. Kawamura, N. Sasaki, T. Iwai, M. Nakano, M. Takagi. Three-dimensional CMOS ICs fabricated by using beam recrystallization. IEEE Electron Device Lett., Vol. EDL-4, Oct. 1983, pp. 366 – 368.

[9] K. Banerjee, S. J. Souri, P. Kapur, K.C. Saraswat. 3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration. Proceedings of the IEEE, Vol. 89, 2001, pp. 602 – 633.

[10] V. Subramanian, M. Toita, N. R. Ibrahim, S. J. Souri, K. C. Saraswat. Low-leakage Germanium-seeded laterally-crystallized single-grain 100 nm TFTs for vertical integration applications. IEEE Electron Device Lett., Vol. 20, Jul. 1999, pp. 341 – 343.

[11] G. W. Neudeck, S. Pae, J. P. Denton, T. Su. Multiple layers of silicon-on-insulator for nanostructure devices. J. Vac. Sci. Technol. B, Vol. 17, no. 3, 1999, pp. 994 – 998.

[12] A. Heya, A. Masuda, H. Matsumura. Low-temperature crystallization of morphous silicon using atomic hydrogen generated by catalytic reaction on heated tungsten. Appl. Phys. Lett., Vol. 74, No.15, 1999, pp. 2143 – 2145.

[13] T. H. Lee. A vertical leap for microchips. Scientific American, Jan. 2002. [Online]. Available: http://www.sciam.com/article.cfm?articleID = 000BD05C-D352-1C6A-84A9809EC588EF21&sc = I100322.

[14] M. Chan. The potential and realization of multi-layers three-dimensional integrated circuit. Int'l Solid-State and Integrated-Circuit Technology, 2001, 40 – 45.

[15] S. A. Kuhn, M. B. Kleiner, P. Ramm, W. Weber. Performance modeling of the interconnect structure of a three-dimensional integrated RISC processor/cache system. IEEE Trans. On Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, Vol. 19, Nov. 1996, pp. 719 – 727.

[16] S. A. Kuhn, M. B. Kleiner, P. Ramm and W. Weber. Performance improvement of the memory hierarchy of RISC-systems by application of 3-D technology. IEEE Trans. On Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, Vol. 19, No. 4, Nov. 1996, pp. 709 – 718.

[17] K. W. Lee, T. Nakamura, T. Ono, Y. Yamada, T. Mizukusa, H. Hashmoto, K. T. Park, H. Kurino, M. Koyanagi. Three-dimensional shared memory fabricated using wafer stacking technology. In: Proc. Int'l Electronic Device Meeting, 2000, pp. 165 – 168.

[18] M. Koyanagi, H. Kurino, Kang Wook Lee, K. Sakuma, N. Miyakawa, H. Itani. Future system-on-silicon LSI chips. IEEE Micro, Vol. 18, July-Aug. 1998, pp. 17 – 22.

[19] S. Strickland, E. Ergin, D. R. Kaeli, P. Zavracky. VLSI design in the 3rd dimension. Integration: the VLSI Journal, Vol. 25(1), Sep. 1998, pp. 1 – 16.

[20] S. M. Alam, D. E. Troxel, C. V. Thompson. A comprehensive layout methodology and layout-specific circuit analyses for three-dimensional integrated circuits. In: Proc. Int'l Sym'm on Quality Electronic Design, 2002, pp. 246 – 251.

[21] J. -Q Lu, et al.. A wafer-scale 3-D IC technology platform using dielectric bonding glues and copper damascene patterned inter-wafer interconnects. In: Proc. Int'l Interconnect Technology Conf., 2002, pp. 78 – 80.

[22] H. B. Pogge. The next chip challenge: effective methods for viable mixed technology SoCs. In: Proc. Design Automation Conf., 2002, pp. 84 – 87.

[23] K. W. Guarini, et al.. Electrical integrity of state-of-the-art 0.13 mm SOI CMOS devices and circuits transferred for three-dimensional (3-D) integrated circuit (IC) fabrication. In: Proc. Int'l Electronic Device Meeting, 2002, pp. 943 – 945.

[24] M. Ieong, et al.. Three dimensional CMOS devices and integrated circuits. In: Proc. Custom Integrated Circuits Conf., 2003, pp. 207 – 213.

[25] B. Burari. Bridging the gap between the digital and real worlds: the expanding role of analog interface technologies. In: Proc. Solid-State Circuits Conference, 2003, pp. 30 – 35.

[26] J. Mayega. 3-D direct vertical interconnect microprocessors test vehicle. In: Proc. Great Lakes Symposium on VLSI, 2003, pp. 141 – 146.

[27] T. Mimura et al.. System module: a new chip-to-chip module technology. In: Proc. Custom Integrated Circuits Conf., 1997, pp. 439 – 442.

[28] Matrix Semiconductor. 3-D technology. [online]. Available: http://www.matrixsemi.com/index.shtml.

[29] IC Knowledge. 2003 Unprobed Wafer Cost. [Online]. Available: http://www.icknowledge.com.

[30] W. Maly, et al.. Smart-substrate multichip-module systems. IEEE Design & Test of Computers, Vol. 11, Summer 1994, pp. 64 – 73.

[31] J. Khare, H. T. Heineken, M. d'Abreu. Cost trade-offs in system on chip designs. In: Proc.13th Conf. On VLSI Design, 2000, pp. 178 – 184.

[32] D. Ammann, et al.. CostAS — KGD process cost modeling. In: Proc. IEEE Innovative Systems in Silicon Conf., Oct. 1996.

3 Design Case Studies

Yangdong (Steve) Deng

Institute of Microelectronics, Tsinghua University
Beijing 100084, P. R. China, dengyd@tsinghua.edu.cn

Wojciech P. Maly

Electrical and Computer Engineering Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891 USA, maly@ece.cmu.edu

Abstract 4 design case studies using the 2.5-D integration scheme are presented in this chapter. The first 3 designs, a crossbar circuit, a Rambus DRAM, and a reconfigurable data-path (PipeRench), are re-designed by exploiting fine-grain inter-chip interconnects. The 4th design study involves a 3-D stacked CPU/memory system. The above design cases studies validate the potential of the 2.5-D integration paradigm from a performance point of view.

Keywords 2.5-D integration, crossbar, Rambus DRAM, reconfigurable data-path, microprocessor, memory, latency.

In Chapter 2, we have proven that the 2.5-D integration scheme offers potential for significant cost saving. Nevertheless, the feasibility of 2.5-D integration can hardly

be fully justified without performance advantages. Beginning from this chapter, we will focus on evaluate the performance potential of 2.5-D integration. In this chapter, we first address this problem by applying the 2.5-D concept to a series of design cases studies.

The designs under investigation are typical VLSI applications but each of them has a quite unique internal organization. Specifically, we will explore three important design attributes: geometrical characteristics of layout, timing performance, and system level throughput. The need for such analysis has been dictated by the fact that no prior design experiences exist to help us appreciate the potential and recognize major limitations of the 2.5-D integration strategy.

3.1 Crossbar

Crossbar is a circuit element used as a switching network, which can be configured to connect any input channel to any output channel. It is widely used in VLSI designs where a number of processing nodes (e.g., ALU or CPU) need to exchange data among themselves or access data from multiple memory blocks. In fact, today the throughput of many high performance applications, such as media processor[1] and network processor[2], heavily depends on the performance of crossbar.

Figure 3.1 shows the stick diagram of a 4×4 crossbar with pass-gates as switches. In the 2-D layout, input terminals are placed on the left border and routed through horizontal wires on the first metal layer. The input wires are connected to the sources of pass-gates. Vertical output wires join to output terminals located on the bottom. Selection signals are routed in poly “zigzagging” through the layout.

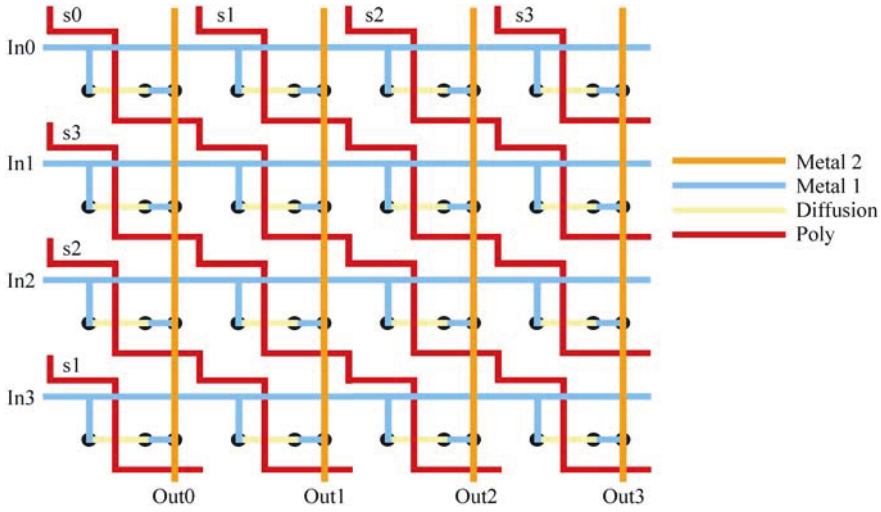


Figure 3.1 Stick diagram of a monolithic crossbar (*see colour plate*)

The main problem associated the 2-D crossbar implementation is its poor scalability. For an $N \times N$ crossbar, the layout area increases with $O(N^2)$, while the wire length of the worst-case signal path rises with $O(N)$. In the worst case, an input signal has to travel all the way from the upper left corner to the bottom left corner in order to reach the output. It should be pointed out that both the input and output lines are heavily loaded due to the large number of fan-outs and fan-ins. For a $N \times N$ crossbar with b bits in each I/O channel, the width and height of crossbar layout are both $4Nbp$, where p is wire pitch (we assume poly, metal 1 and metal 2 have the same pitch). If we assume $N = 32$, $b = 64$, and $p = 0.4$ mm (for 0.18 μm process), the total crossbar area would be 13.1 mm^2 and the worst-case interconnection length a signal needs to travel is 6.6 mm.

With the 2.5-D stacking technology, we are able to construct a more efficient packing for the crossbar circuit. In the 2.5-D layout with two stacked chips, each chip will route half input and output wires as shown in Fig. 3.2. Now every source region (connected to an input line) is shared by two abutted pass-gates. In

every pair of abutted pass-gates, one drain node is connected to an output wire on the same chip, while the other drain is connected to another output wire laid on the other chip through an inter-chip contact. For the example shown in Fig. 3.2, if s0 is high, input in0 will drive out0. If s2 is high, in0 drives out2 through an inter-chip contact. One possible problem is that there may not be enough space to place all inter-chip contacts directly above the drain regions. If this happens, we can use additional wires and metal-to-metal vias to properly spread the inter-chip contacts.

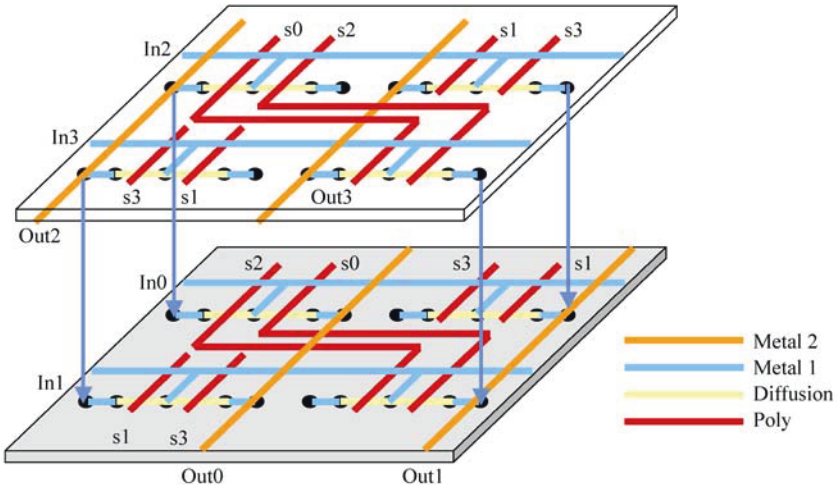


Figure 3.2 Stick diagram of a 2.5-D crossbar (*see colour plate*)

Under this 2.5-D layout model, the width of the crossbar layout in one chip level is *3Nbp* and the height is *2Nbp*. For the above example ($N = 32$, $b = 64$, and $p = 0.4$ mm), the total area will be 9.8 mm^2 (sum of two chips) and the worst-case interconnection length a signal needs to travel is 4.1 mm. In other words, potentially the layout area can be reduced by 25% and worst-case interconnection length can be reduced by 37.5%. This case study, although simple, evidently demonstrates the flexibility of the 2.5-D layout.

3.2 A 2.5-D Rambus DRAM Architecture

Rambus DRAM, or RDRAM, is a high-speed memory for high-performance microprocessor and graphic applications[3−6]. It is featured by a large number of internal banks and a narrow (16-bit data) but extremely high-speed bus. The multi-bank architecture allows a sustainable high bandwidth for multiple, random memory transactions. The narrow bus interface is aggressively optimized for performance and can be accessed on both clock edges. In this section, we introduce two enhancements to the RDRAM architecture using the 2.5-D concept.

3.2.1 Tackle the Long Bus Wire

Figure 3.3 illustrates the internal organization of a typical 128 MB Rambus DRAM chip. The memory is organized into 32 banks with each bank holding 4 MB data.

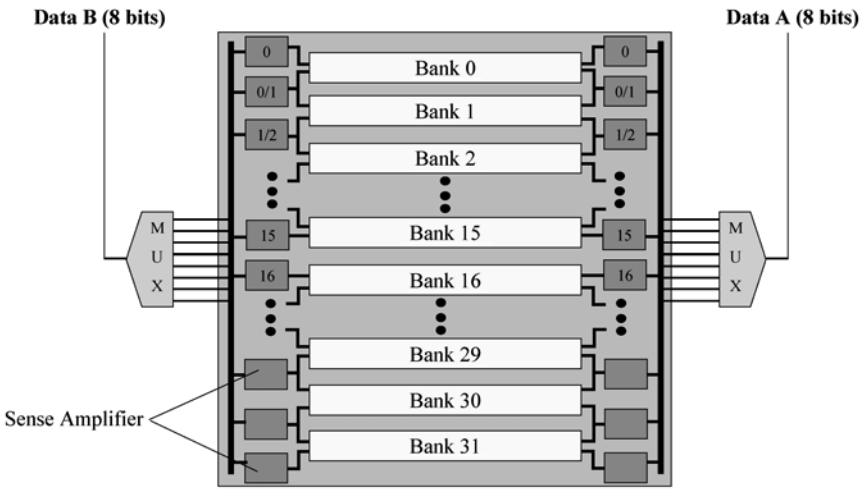


Figure 3.3 Rambus DRAM

A bank is actually a DRAM array composed of 1024 rows of 256 “dualocts”. A dualoct contains 16 bytes and is the smallest unit of memory that can be addressed in a RDRAM. Since neighboring banks could share a group of sense amplifiers, only 16 out of the 32 banks can be active at the same time.

During a read operation, a 16-byte dualoct from a bank will be selected. Then half (8 bytes) of the dualoct flows to the internal bus one the left side of the bank and the other half flows to the internal bus on the right. The 8-byte wide internal bus on the right (left) is connected to the data A (B) bus through an 8∶1 multiplexor and at every clock edge (rising or falling) one byte of data is put onto A (B). This way data A and B buses together can output two bytes at every clock edge.

Clearly, the long internal buses account for a considerable percentage of cycle time. We used the UCLA’s IPEM tool[7] to estimate the delay of the bus wire assuming a 0.13 μm DRAM process. For a 1 Gb RDRAM chip with a die area of 203.6 mm^2 (~2 mm × 1 mm) and a working frequency of 1.3 GHz as reported by ICKnowledge[8], the internal bus wire has a length of 20 mm. When properly buffered, the bus wire will have a delay of 1.23 ns. As we know, 8 bytes of data loaded to the internal bus and then read out in 8 cycles. Hence, the internal bus delay should be 6.00 ns. If 3-D stacking technology is viable, we can “fold” the DRAM into two chips and stack them together in a way like Fig. 3.4. Placing the bus wires on one chip, the cells can be assigned to another chip connecting to the bus through inter-chip contacts. As a result, a bus wire will have a delay of 0.61 ns, which implies the clock frequency can be improved to 1.6 GHz, or a 23.0% improvement in bandwidth. If we stack the DRAM into three layers, the bus wire delay is 0.41 ns and the corresponding working frequency is 2.4 GHz.

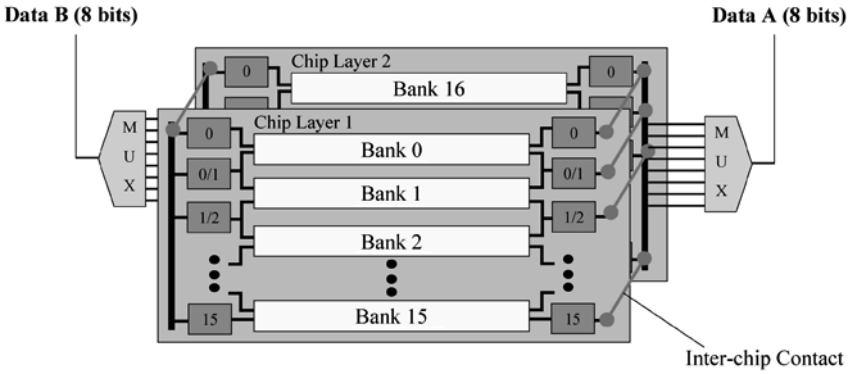


Figure 3.4 2.5-D Rambus DRAM

3.2.2 Serialized Channel in the 3rd Dimension

One RDRAM data channel is fixed at a width of 16 bits and thus each RDRAM chip can independently feed a channel. To build a memory system, several RDRAM chips will be serially connected, in a manner illustrated in Fig. 3.5[9–11]. As a result, in the worst case data has to travel across every chip in the memory system and every inter-chip link interconnecting the memory system. Since physical length of each inter-chip link is of the order of 1 cm, the total length of a signal path in a typical memory system (e.g., 512 MB DRAM consisting of 4 RDRAM chips) will be several centimeters, which will result in a serious delay. The poor scalability is a serious concern when a large number of RDRAM devices have to be integrated. To guarantee a high bus speed, a PCB board carrying RDRAM chips has to be manufactured with a superior quality to reduce noise, stray capacitance and impedance, and all kinds of variations. As a result, the long, fast bus usually implies a high manufacturing cost. Meanwhile, even with high-speed on-board interconnections, the length of the bus and clock wires is only expandable to a limit of 10 cm[11], which gives an upper limit on the DRAM capacity for a system.

If the 2.5-D technology is available, it's possible to re-organize the RDRAM chip so that the signal path can be significantly reduced. Suppose we need to design a memory system composed of four RDRAM chips. In the conventional solution, the four chips will be serially connected as shown in Fig. 3.5. In a 2.5-D stacked memory system illustrated in Fig. 3.6, the DRAM cells can now be placed into four layers and vertically stacked. The memory bus will be through inter-chip contacts, which have only a vertical height of <50 μm. As a result, 2.5-D stacked Rambus DRAM has a considerable potential to achieve superior performance at a relatively low cost.

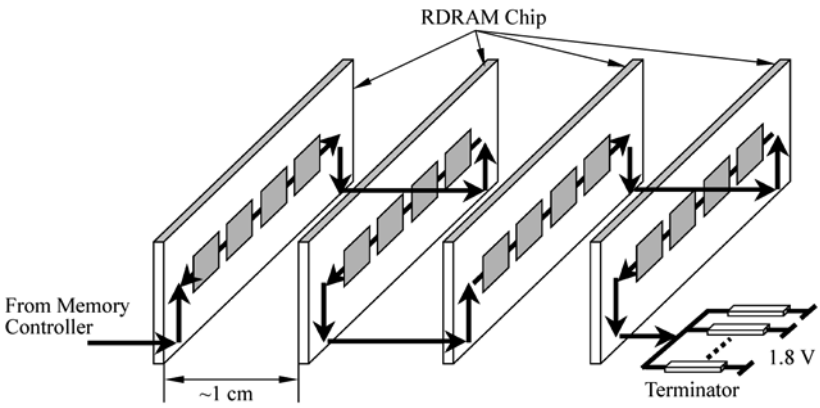


Figure 3.5 RDRAM memory system

Moreover, since the stacked RDRAM is a complete system, it can be built with larger freedom. For instance, a 2.5-D RDRAM can be configured as one single RDRAM channel or multiple channels by properly grouping internal banks in the vertical direction. One such a 4-channel configuration is shown in Fig. 3.6. Here one channel consists of 16 banks placed into four layers with data flowing as the indicated by the arrows.

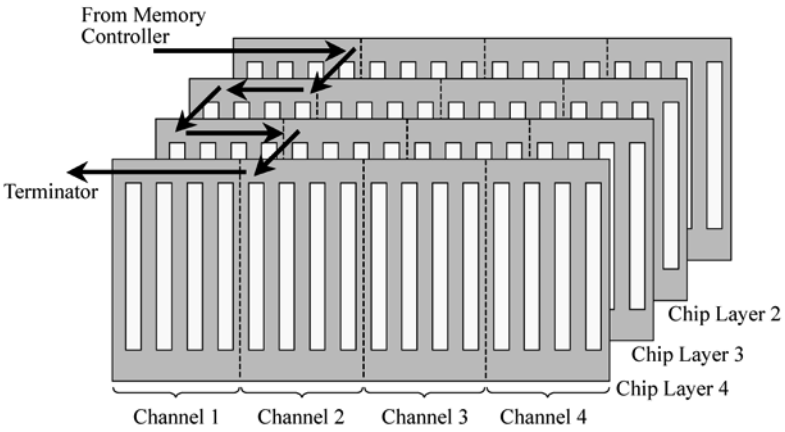


Figure 3.6 3-D Rambus DRAM: 4-channel configuration

From the above analysis, one can see that the 2.5-D integration style can greatly help build faster DRAM memories in two aspects: (1) allow more efficient layout organization, and (2) removing inter-chip bus connecting.

3.3 A 2.5-D Redesign of PipeRench

To assess the performance potential of the 2.5-D integration, we re-design of an existing monolithic system, PipeRench[12], into a prototype 2.5-D system. The PipeRench chip is a re-configurable datapath targeted for multimedia processing. It was originally designed by CMU students and faculty and has been successfully fabricated in ST Microelectronics' 0.18 μm CMOS process. The chip consists of 3.65 million transistors and has a die area of 7.3 mm×7.6 mm. The layout of original PipeRench chip is illustrated in Fig. 3.7. To make the structure of the reconfigurable fabric more visible, we hide metal layers 5 and 6 in the above drawing.

PipeRench is designed as a virtualized programmable datapath for media

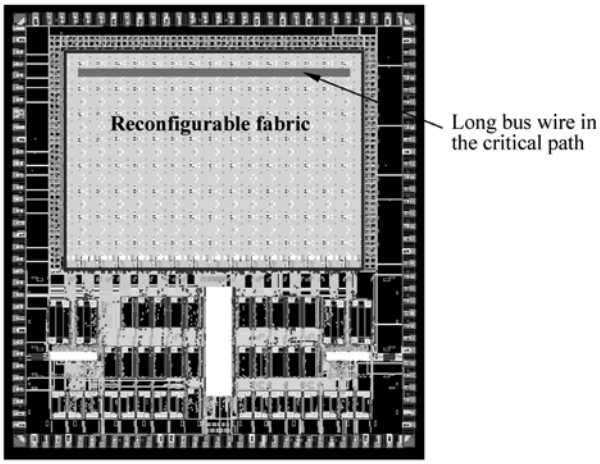


Figure 3.7 Original monolithic implementation of PipeRench

processing applications. It can be dynamically reconfigured into different pipeline structures to better adapt to a target application. In this design, the re-configurability is realized by a reconfigurable fabric, which consumes around 60% of total layout area. The fabric consists of 256 regularly placed processing elements (PEs) evenly divided into 16 stripes (16 PEs in each stripe). Those PEs in one stripe are interconnected by a 128-bit bus. Outputs from a PE are also feed to the registers of corresponding PE in the next stripe. A PE is composed of a configurable ALU composed of eight 3-input look-up tables, a register file and switching logic. By properly setting a set of control bits, a PE can be programmed to implement desired functionality and interconnection pattern.

The critical path of PipeRench chip lies completely within one stripe. The signal path is shown in Fig. 3.8. In fact, because one stripe can potentially be configured as one single entity to perform certain computation, the critical path has to account for the delay from the rightmost PE to the leftmost PE in the same stripe. In Fig. 3.2 the critical path is sketched in bold line. In the worst case,

signals stored in the register file of rightmost PE can be required by the functional unit of the leftmost PE. After computation, the result will be stored in the register file of leftmost PE. Under such a situation, signals have to be read from the rightmost PE and then delivered through the intra-stripe bus wires. Intra-stripe bus is implemented in 4th metal layer and driven by a large buffer (32X driving strength). Since the intra-stripe bus has to span almost the dimension of the chip, it thus has a length of around 7 mm. It takes about 4 ns for a signal to be transited through the bus, while the critical path has a delay of around 8 ns. Obviously, the intra-stripe bus poses a bottleneck to the whole system. However, it is inevitable in the two-dimensional integration of the programmable fabric.

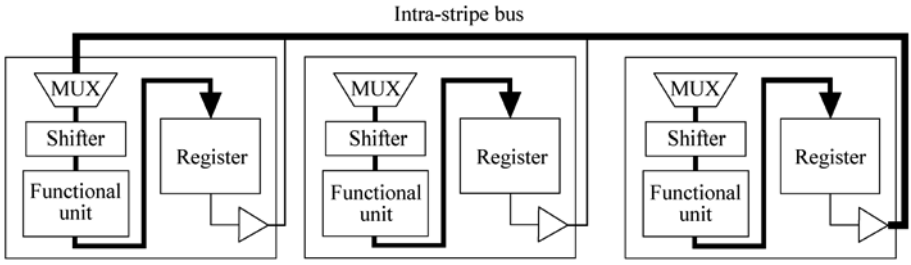


Figure 3.8 Critical path of PipeRench system

3.3.1 The 2.5-D Implementation

According to the analysis presented in the previous section, we realized that the most effective way to reduce critical path delay is to fold the fabric into two stacked chips under the 2.5-D scenario. Therefore, we vertically split the chip into two halves and the resultant two chips can be bonded in a face-to-face manner. The layouts of the two chips in 2.5-D system are shown in Fig. 3.9. Note that the size

of inter-chip contacts has been exaggerated in Fig. 3.9.

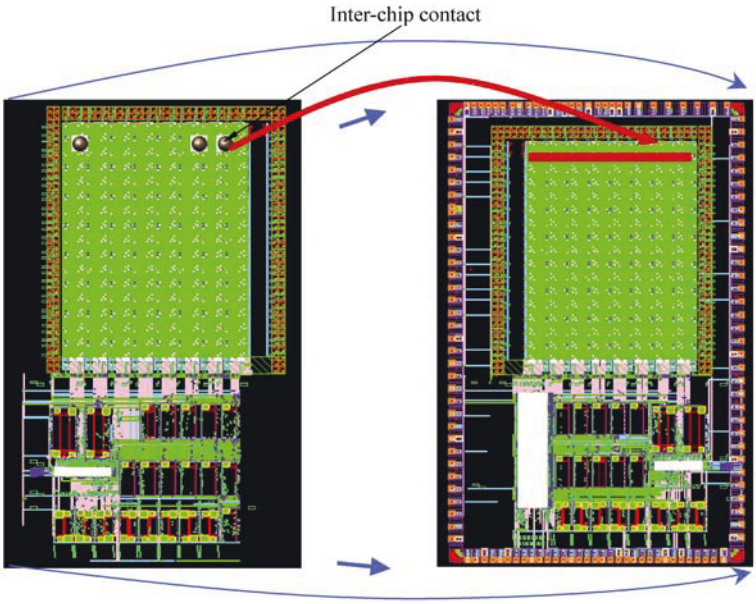


Figure 3.9 The 2.5-D re-design of PipeRench (*see colour plate*)

We manually adjusted the following layout structures to split the chip into two halves. The aforementioned intra-stripe bus wires are placed on the bottom chip, while processing elements assigned the top chip are connected to the bus through inter-chip contacts. Thus the wire length of such a bus wire is only one half of the original wire length in the monolithic design. Totally we use 16384 inter-chip contacts for the bus connection.

All the pads are placed on the bottom level chip. Since the bottom chip has a smaller footprint than the original design does, the distance between pads is uniformly scaled down so that the pads can be fitted to the short perimeter.

In the original monolithic design, the reconfigurable fabric is surrounded by a

power ring and a ground ring. We implemented such rings with proper width to fit the two halves of the reconfigurable fabric.

In the new design, we use two H-trees to deliver clock signals for the top level chip and bottom level chip, respectively. The two H-trees are interconnected through 128 inter-chip contacts to reduce clock skew.

Finally, we need 900 inter-chip contacts for power distribution so make sure: (1) current density of a 1-micron thick Al power wire should be within 0.4 mA/micron to 1 mA/micron[13]; (2) current density of an inter-chip contact should be under 0.1 mA/micron[13]. According to our calculation, the worst-case IR dropping is within 9.6 mV.

In the 2.5-D layout, the bottom chip has a surface area of 3.9 mm by 7.8 mm. If we assume the area pitch of inter-chip contact is 5 μm × 5 μm, we can have up to 1.2 million inter-chip contacts on the chip surface. Since the 2.5-D re-design is done manually, the actual number of used inter-chip contacts is far less than the above available number.

3.3.2 Simulation Results

We built SPICE models for the critical paths in both implementations. The parametric RC parameters for in-chip layout structure are extracted using a commercial tool[14]. Obviously, full-chip SPICE simulation would be infeasible. Accordingly, we built a partial layout specifically for RC parameter extraction. The partial layout includes the original logic circuits, driver buffer, bus wire belong to a specific critical path as well as 3 neighboring wires on both side and typical layout structures in adjacent metal layers.

Generally, the inter-chip contact should be modeled with a distributed RC model. However, we do not know the exact structure of inter-chip contact and we instead use a PI-model to represent the RC effect of inter-chip contacts on the bus wire. We found that, when in the range of 100 – 10 k Ohm, the resistance value in the PI-model does not have a significant effect on the delay. We assume the two capacitors in the p-model have the same value.

Table 3.1 SPICE simulation on the critical path

Lumped Capacitance of inter-chip contact (fF)	Bus Wire Delay (ns)	Critical Path Delay (ns)	Clock Frequency (MHz)	Improvement to 2D Solution
0.1	1.02	5.32	188.0	55.1%
1	1.11	5.41	184.8	52.5%
15	1.50	5.80	172.4	42.2%
65	1.95	6.25	160.0	32.0%

Table 3.1 shows the simulation results with capacitance values ranging from 0.1 fF to 215 fF. It has been reported that the micro-bump bonding technology with a 30 micron pitch has a capacitance of only 10 fF and no inductance component up to 90 GHz[15]. We tried a wide range of value to observe the general trend. The results show that the potential speedup in 2.5-D system is considerable: When the lumped capacitance of inter-chip contact is 15 fF (roughly corresponds to the lumped capacitance of a 75 μm long metal wire in 0.18 μm process), we can reduce the critical path delay from 8.3 ns to 5.8 ns, corresponding to a 42% speedup in clock frequency.

From the design experience of 2.5-D PipeRench, we found that substantial performance gain can be achieved because the 2.5-D integration style provides significant flexibility to cut down long wires through the usage of inter-chip contacts.

3.4 A 2.5-D Integrated Microprocessor System

In Chapter 1, we discussed the performance gap between microprocessors and DRAM based main memory. The so-called "memory wall" affects microprocessor performance in two aspects: bandwidth and latency. Latency is the waiting time after a memory request is sent to the memory controller, while bandwidth is the maximum throughput a memory system could provide.

With the introduction of advanced DRAM architectures such as Rambus[3] and DDR Ⅰ/Ⅱ[9], the demand for memory bandwidth has been mitigated to a certain extent. However, the number of available I/O pins still poses a limit on the performance of such systems as like network processors and graphic processing units, which interface with the outside through memories. For instance, let's consider a network processor handling OC768 packets with a data rate of 40 Gbits per second (bps). Since every packet needs four memory accesses per packet and usually a network processor is in charge of duplex communication, the memory requirement will be $2 \times 4 \times 40\ \mathrm{G} = 320$ Gbps. Assuming effective memory bandwidth is ~60% of peak bandwidth, the required memory bandwidth is 500 Gbps. For off-chip SDRAM running at 266 MHz, around 2000 pins will be needed to satisfy the bandwidth requirement.

Compared with bandwidth, memory latency is improved at an even slower pace. The classical way to hide the excessive latency is to place frequently used

instructions and data in caches with a smaller capacity but a shorter latency. Today's microprocessors usually exploit 2 to 4 levels of cache memories on top of the main memory. However, different applications have varying cache behaviors. According to our detailed instruction simulations by SimpleScalar[16] on SPEC2000 benchmarks[17], the miss rates of level 2 cache could vary by two magnitudes. Meanwhile, server applications usually have much lower cache hits ratio[18]. As a result, memory latency is usually the main performance bottleneck.

With the development of 2.5-D stacking technology, it's now feasible to build the logic circuits and DRAM main memory on separate dies and bond them together in the vertical direction. The memory bus will be through inter-chip contacts that can be placed on chip surfaces. By replacing off-chip interconnections, memory latency could be significantly improved. Meanwhile, it is possible to deploy very wide memory bus to enhance the bandwidth between memory and CPU core (literally thousands of signals even in a 1 mm × 1 mm die surface). In this section, we will explore the design issues for a 3-D integrated microprocessor/DRAM system.

3.4.1 A 2.5-D Integrated Microprocessor System

We chose a scaled version of the HP-Compaq Alpha 21364 processor[19] in this research because of the availability of the design details and corresponding development tools (i.e. instruction simulator, cross compiler, and so on). The latest version of Alpha 21364 is built in a 0.18 μm process and has a die area of 397 mm^2. We scaled the floorplan to a 0.13 μm process and the scaled version has a die area of 208 mm^2. Figure 3.10 shows the floorplan of Alpha 21364 (the

chip dimension has been scaled to a 0.13 μm process). The target microarchitecture is an aggressive 8-way, out-of-order, superscalar microprocessor running at 4 GHz. It accepts Alpha instruction set. The internal configuration of the processor listed in Table 3.2.

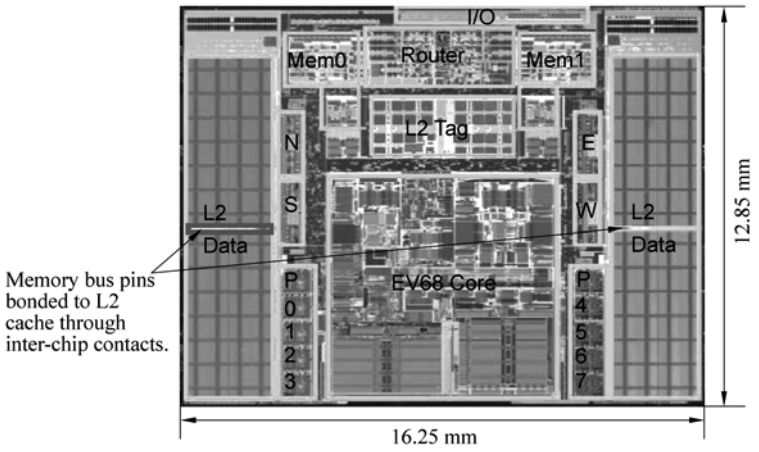


Figure 3.10 Alpha 21364 floorplan and memory bus placement

Table 3.2 Configuration of target microprocessor

Parameter	Configuration
General	Out-of-order microprocessor with Alpha instruction set
Clock Frequency	4 GHz
Issue Width	8 instructions/cycle
Decoder Width	8 instructions/cycle
Commit Width	8 instructions/cycle
L1 Instruction Cache	64 KB
L1 Instruction Cache Latency	1 cycle

(Continued)

Parameter	Configuration
L1 Data Cache	64 KB
L1 Data Cache Latency	1 cycle
Unified L2 Cache	2 MB
L2 Cache Latency	12 cycles
Off-chip Memory Access Latency	400 cycles
Memory Access Cycle	4 cycles
# Integer ALUs	4
# Integer Multiplier/Dividers	2
# Float ALUs	2
# Float Multiplier/Dividers	1

The DRAM is also assumed to be built in a 0.13 μm process. The memory bus between L2 cache and DRAM is placed in the middle of the L2 cache, which is marked as red rectangles in Fig. 3.10. The DRAM will be placed on the top of the microprocessor in a way illustrated by Fig. 3.11. As for the main memory, we selected a high-end, Rambus DRAM with a clock of 1 GHz. Since the CPU clock has a frequency of 4 GHz, the access cycle time of the main memory is 4 CPU cycles. In the remaining part of this section, the word “cycle” always refers to a CPU cycle. On the other hand, the memory latency value is usually determined by the specific machine configuration and typically values are in the range of 100 cycles to 500 cycles (e.g., [20]). Because our target microprocessor is very aggressively clocked, we assume a memory latency of 400 cycles.

As illustrated in Fig. 3.12, a microprocessor interfaces with the main memory

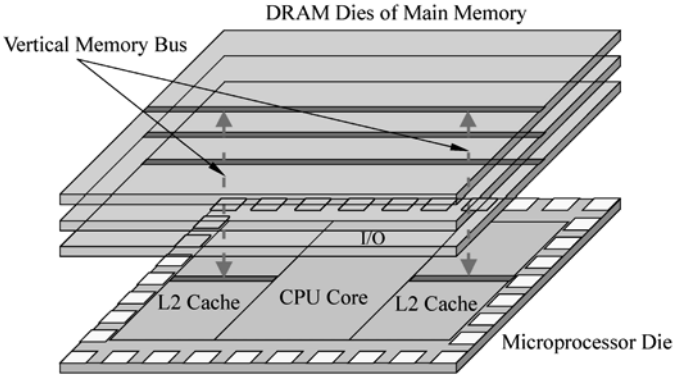


Figure 3.11 A 2.5-D stacked microprocessor and DRAM

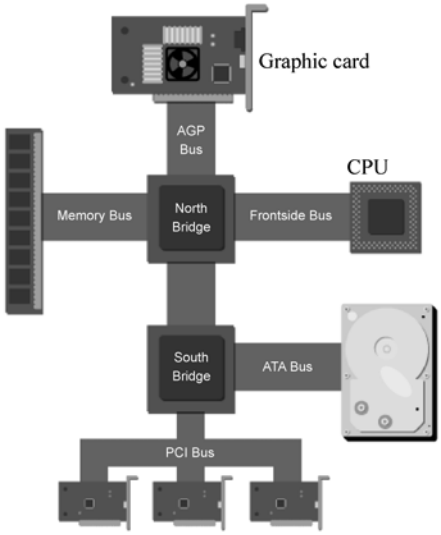


Figure 3.12 A diagram of computer system

through a memory controller (often referred to as Northbridge in the computer industry), which is traditionally a separate chip in the motherboard terminology. A recent trend is to integrate one or more memory controllers onto the microprocessor chip (e.g., the Alpha 21364 microprocessor integrates two on-chip

memory controllers). The memory traffic has to be through on-board wires and off-chip I/O pads before reaching the microprocessor. Therefore the memory latency consists of three components: the latency due to the memory chip itself, the chip-to-chip latency, and the memory controller delay. Typically it can be assumed that the combined chip-to-chip latency constitutes 1/3 to 1/2 of the total latency[21,22].

With the stacked DRAM, the off-chip delay can be almost completely removed since no off-chip wires and I/O pads are necessary. Of course, it still too early to use an exact latency value for the stacked DRAM at the present due to the inviolability of technology parameters. Assuming a 400-cycle off-chip memory latency, we set the upper bound latency (pessimistic estimation) of stacked DRAM is 300 cycles, while a lower bound latency (optimistic estimation) is 200 cycles.

Today a typical off-the-shelf 1 G-bit DRAM in a 0.13 μm process has a die size of 204 mm^2[8]. Such a DRAM die can hold 192 MB if we assume it would have the same die size as the microprocessor. Certainly this capacity is too restricted with today's standard. To have a larger capacity, we could consider the following three options:

Stacking more layers of DRAMs on the top For instance, when 3 layers of DRAM dies are stacked on the top of the microprocessor die, 576 MB is available, which is applicable for most applications. A 5-layer DRAM could suffice high-end applications. The major concern of such a configuration is heat dissipation. Fortunately, DRAM usually has relatively low power consumption. If there's still excessive heat build-up, hardware based heat dissipation techniques like heat pipes (e.g., dummy inter-chip interconnects) have to be installed.

Stacking a larger DRAM die Having a larger die provides higher memory capacity, but certainly requires a bigger footprint than that of the bottom layer microprocessor. The over-sized DRAM dies may pose difficulty to the package.

Installing extra off-chip main memory The stacked DRAM and off-chip DRAM have different access delays and thus this approach implies a Non-Uniform Memory Architecture (NUMA)[23]. The operating system has to take charge of the resultant memory management issues.

3.4.2 An Analytical Performance Model

Modern microarchitecture designs heavily depend on cycle-accurate instruction simulators like SimpleScalar[16]. Such simulators enable architects to evaluate performance of a future architecture on existing computer platforms. The simulation process, however, tends to be very time-consuming. It's typically 100 times slower than the native execution. For instance, it usually takes days or even weeks for the SimpleScalar simulator to finish running programs from the SPEC2000 benchmark[17] and SPEC95 benchmark[24] suites. To fast investigate the performance implications of the 2.5-D integrated microprocessor system, we resort to an analytical performance model based on the work by Matick et al.[25].

The most commonly used metric for CPU performance is Instruction per Cycle (IPC). However, its reciprocal, Cycle per Instruction (CPI), is a more appropriate metric for an analytical analysis since it could be formulated as a sum of factors representing the impact of various microarchitecture features. The essential idea of the analytical modeling is illustrated in Fig. 3.13.

On an ideal microprocessor with an infinite Level 1 (L1) cache, every memory access hits. The resultant CPI value, CPI_{ideal}, is completely determined by the inherent parallelism of the application as well as the limit of computational resources. Certainly there is no such an ideal microprocessor in reality. As a matter of fact, L1 cache has to be configured with a relatively small capacity so

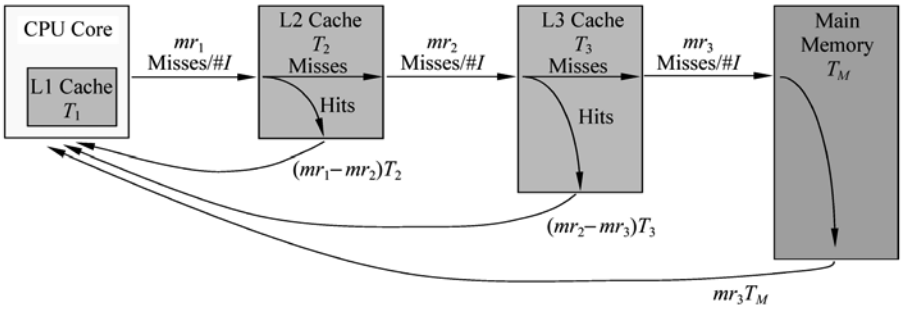


Figure 3.13 CPI calculation

that it's can be accessed within one cycle (at most two cycles). Consequently, a given percentage of all memory accesses would miss the L1 cache and has to be directed to Level 2 (L2) cache. The performance of L1 cache is reflected by its miss rate, mr_1, which is calculated as:

$$mr_1 = \#\text{misses}_1 / \#\text{instructions} \tag{3.1}$$

Similarly, L2 cache is characterized by mr_2:

$$mr_2 = \#\text{misses}_2 / \#\text{instructions} \tag{3.2}$$

As long as a L1 cache miss happens, CPU will then query the L2 cache as illustrated in Fig. 3.13. Every access to L2 cache, no matter a hit or a miss, has to incur a latency time, T_2, which is the sum of such factors as SRAM pre-charge delay, word line decoding, bus delay, and so on. The latency incurs a penalty to the CPI metric as:

$$P_{C2} = (mr_1 - mr_2)T_2 \tag{3.3}$$

Besides the memory word required by the current instruction, a L1 cache miss

results in data exchange between L1 and L2 caches. First of all, memory words at the adjacent addresses have to be filled into the L1 cache. The number of words depends on the size of a cache line. If the cache line contains dirty data, i.e. memory words recently modified in L1 cache but not yet written to L2 cache, the data has to be written out to L2 cache. Besides, the stale L1 data also needs to be replaced if it has not been accessed for a given period. This effect is called cast out and it can be assumed to happen at a rate of 1/3 mr_1.

All these data traffics are through a dedicated bus between L1 and L2 caches. Accordingly, the bus has to be modeled as a queue. The service time S_2 is constant since every time a fix-sized cache line is transferred:

$$S_2 = T_{B2} \times (\#\text{Bytes}/\text{LineSize}) \times (\#\text{Bytes}/\text{BusWidth}) \tag{3.4}$$

The average delay in a queue is composed of two components: waiting time Q_{w2} and service time S_2. The waiting time can be formulated a function of bus utilization ratio, U_2:

$$Q_{w2} = 0.5 \times \frac{U_2}{1 - U_2} \tag{3.5}$$

Given a CPI value, the utilization rate can be calculated as:

$$U_2 = (mr_2/\text{CPI}) \times S_2 \tag{3.6}$$

This way the penalty caused by the L1-L2 bus can be formulated as:

$$P_{B2} = (mr_1 - mr_2) \times (Q_{w2} + S_2) \tag{3.7}$$

Considering the whole memory hierarchy with all misses will finally hit the

main memory, CPI value can be analytically expressed as:

$$\mathrm{CPI} = \mathrm{CPI}_{\mathrm{ideal}} + \sum (P_{C2} + P_{B2}) \tag{3.8}$$

Of course, the CPI value is to be determined yet and thus our model is not close-formed. As a result, we calculate CPI in an iterative manner. Beginning with the ideal CPI, a utilization rate for each bus can be derived and then the penalty is added to the current CPI value. The updated CPI value can be similarly used to calculate updated utilization rates. This process is iterated until a convergence criterion is satisfied.

With the above model, we could derive the distribution of CPI metric with regard to the L2 cache miss rate and the main memory latency. The memory latency is measured in the number of CPU clock cycles.

The distribution graph shown in Fig. 3.14 is derived by assuming an ideal CPI

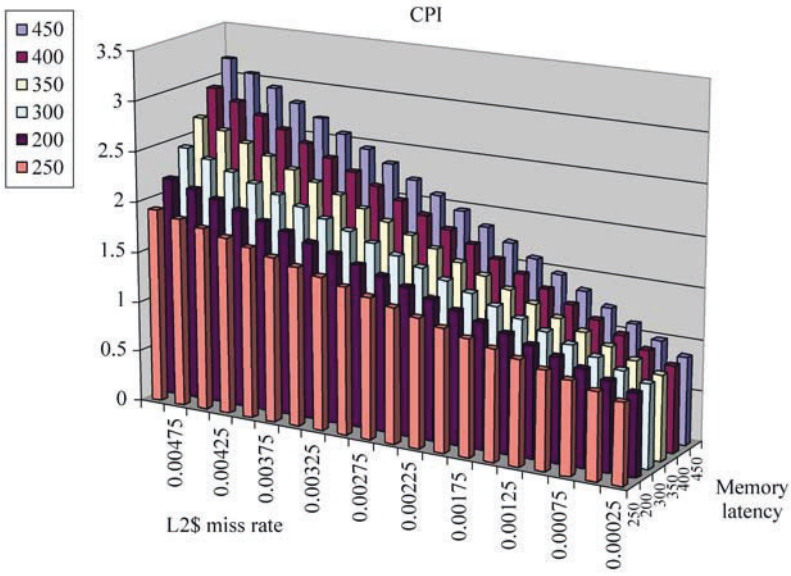


Figure 3.14 CPI with regard to main memory latency and L2 cache miss rate (*see colour plate*)

of 0.5 and a constant L1 cache miss rate of 0.02. When the L2 cache miss rate is relatively low (say, <0.001), the CPI value is not very sensitive to the memory latency. As a matter of fact, when the memory latency changes from a typical value of 450 cycles to a very optimistic value of 200 cycles, the CPI value only improves by 7% with a L2 cache miss rate of 0.00025. However, with relatively higher L2 cache miss rates, the CPI value is expected to have a dramatic improvement. For instance, with a L2 miss rate of 0.003, the CPI value increases by 34% when the memory latency reduces from 450 to 200 cycles. The above results clearly demonstrate the potential of the 2.5-D integration (corresponding to memory latency values below 300 cycles).

3.4.3 Detailed Performance Simulation for Reduced Memory Latency

The analytical model derived in the previous section is actually a first order CPU performance model. It provides important intuitions on choosing programs that could best benefit from the stacking of DRAM, although it ignores many other second order microarchitecture features (e.g., translation-look-aside buffer). We then performed cycle accurate cache simulations on the SPEC2000 benchmark suite using the cache simulator, Sim-Cache, from the SimpleScalar toolset. According to the cache simulation results, we could pick up those SPEC2000 benchmarks that have L2 cache miss rate higher than 0.1%. The characteristics of these benchmarks are shown in Table 3.3.

We then performed detailed instruction simulation on the benchmarks listed in Table 3.3. We differentiate three configurations: (1) a baseline 2-D microprocessor with off-chip main memory (Memory latency = 400 cycles), (2) a 2.5-D integrated

Table 3.3 SPEC2000 benchmark programs under study

Benchmark	L2 Cache Miss Rate	Category	Memory Footprint (MB)
equake	0.0031	Seismic Wave Propagation Simulation	50
swim	0.0379	Shallow Water Modeling	194
art	0.0195	Image Recognition / Neural Networks	4
applu	0.0027	Parabolic / Elliptic Partial Differential Equations	181
lucas	0.053	Number Theory / Primality Testing	146
vpr	0.0026	FPGA circuit Placement and Routing	45
mcf	0.0159	Combinatorial Optimization	190
facerec	0.0013	Image Processing: Face Recognition	17
fma3-D	0.0015	Finite-element Crash Simulation	103

microprocessor with memory latency of 300 cycles, which can be seen as a pessimistic estimation on the performance of the stacked DRAM, and (3) a 2.5-D stacked microprocessor with memory latency of 200 cycles. The speedup values relative to the baseline configuration are illustrated in Fig. 3.15, while the raw data is shown in Table 3.4. The performance speedup is quite consistent except *lucas*, which seems to be more resource limited due to its intense numerical computation. On average, even with the upper bound memory latency (300 cycles), a 23.7% improvement can be achieved; while with the lower bound memory latency (200 cycles), 81.4% speedup can be attained. The performance advantage is especially salient on three benchmarks, *swim*, *applu*, and *mcf*, where the IPC values are more than doubled.

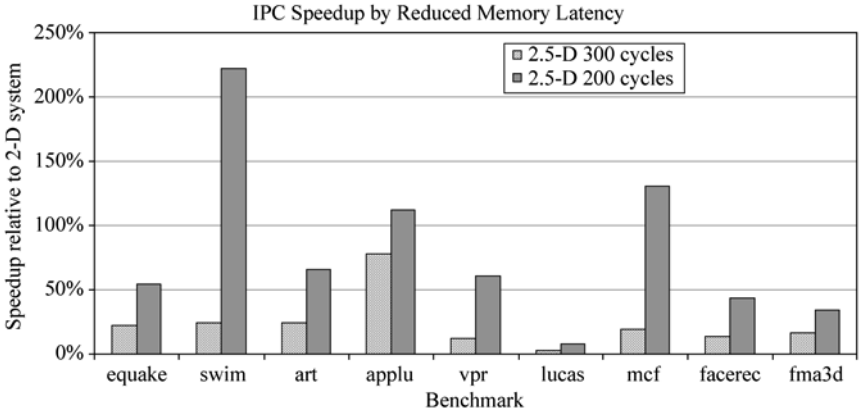


Figure 3.15 IPC Speedup by reduced memory latency

Table 3.4 IPC improvement by Reduced Memory Latency

Bench-mark	L2 Cache Miss Rate	IPC (Baseline 2-D processor)	IPC (3-D Processor Tm = 300)	Speedup relative to Baseline Processor	IPC (3-D Processor Tm = 200)	Speedup relative to Baseline Processor
equake	0.0031	0.7275	0.8910	22.474%	1.1243	54.543%
swim	0.0379	0.8097	1.0091	24.626%	2.6085	222.156%
art	0.0195	0.3017	0.3756	24.495%	0.4989	65.363%
applu	0.0027	1.0661	1.8926	77.526%	2.2649	112.447%
vpr	0.0026	0.4583	0.5131	11.957%	0.7379	61.000%
lucas	0.0053	0.1112	0.1146	3.058%	0.1201	8.004%
mcf	0.0159	0.1562	0.1859	19.000%	0.3608	131.000%
facerec	0.0013	1.7509	1.9908	13.702%	2.5145	43.609%
fma3d	0.0015	1.5153	1.7611	16.221%	2.0394	34.586%
equake	0.0031	0.7275	0.8910	22.474%	1.1243	54.543%

3.5 Observations

In this chapter, we applied the 2.5-D integration concept to 4 design case studies in the custom design style. The first two cases, although relatively simple, demonstrate the potential of the 2.5-D layout to achieve improved interconnection characteristics. The re-design of PipeRench proves the potential of the 2.5-D integration for better timing performance. The last design provides strong evidence for the 2.5-D implementation to get better system level throughput. Clearly, the 2.5-D implementations could deliver superior performance than their monolithic equivalents. To summarize, the advantages include the following:

Improved layout efficiency and interconnect characteristics The extra dimension in 2.5-D layout space often allows designers to find a more flexible and efficient packing for the designed system. A 2.5-D layout can be designed to effectively reduce long wires and thus improve timing performance. The packing flexibility has important implications for interconnect-intensive VLSI applications. For instance, programmable devices like FPGA and CPLD require a significant portion (e.g., 90%) of chip area dedicated to programmable interconnects (i.e. wires, switches, and corresponding memory bits storing the configuration information). The routing resource consumes ~80% of signal path delay and >60% dynamic power. Thus it is extremely advantageous to be able systematically reduce interconnect length. Unlike the monolithic implementation, where all the logic devices and interconnects have to compete for the same substrate, a 2.5-D FPGA could have logic devices, programmable interconnects and configuration memories assigned to three different chips that are stacked in the vertical direction. This way it's possible to reduce interconnect length and maximize the logic density at the same time.

Improved memory latency and bandwidth The 2.5-D paradigm provides an elegant way to boost the throughput of microprocessor systems by stacked memories and vertical memory buses through inter-chip contacts. The performance improvement comes from two sources: reduced memory latency through the removal of off-chip memory bus and increased memory bandwidth by deploying very wide memory bus. In our design case study of stacking DRAM on top of a microprocessor, we have demonstrated the performance potential of the 2.5-D integrated system. From a bandwidth point of view, a wide bus (e.g., more than 512 bits wide) is much desired for memory bandwidth intensive applications like graphic/video processing. Such wide buses could easily be built in 2.5-D ICs, while they have to be infeasible at printed circuit board level. The above advantages would be even more noteworthy for multi-processor systems. In a monolithically integrated multi-processor, the number of processors would increase in proportion to the chip area, while the number of I/O pads only increases in line with the perimeter of the chip. Consequently, multiple processors on the chip would have to share an I/O port and the memory latency and bandwidth issues could only get worse. In 2.5-D ICs, different processors could access the memory banks stacked on their tops in parallel. In addition, the memory banks could be interconnected by a high-speed network allowing multiple accesses at the same time.

Easier hybrid integration The 2.5-D integration scheme is very natural for the integration hybrid technologies. Our case study on the microprocessor only partially demonstrates this advantage. A future research project should focus on the 2.5-D integration of wireless chipsets. Today a typical cell phone contains at least 6 different technologies: Bi-CMOS for the radio transceiver, analog CMOS for the radio and audio codec, digital CMOS for digital baseband processor and application processor, FLASH memory, high voltage CMOS for power management,

and passives such as SAW filters and inductors[26]. With the multi-band, multi-mode phones becoming popular, the complexity of wireless chipsets is still fast increasing. In the foreseeable future, it is unlikely a one-chip solution could deliver the full functionality and performance at a cost-efficient level. On the other hand, reducing volume and weight and lowering power consumption are always of top concern for the cell phone industry. Based on the above two contradicting requirements, a cellular chipsets is naturally suitable to be implemented in the 2.5-D approach. The 2.5-D integration scheme would enable a wireless system to be built in a modular manner, where add-value features like digital cameras and BlueTooth transceivers could simply be fabricated as additional layer of chip and selectively installed.

References

[1] C. Kozyrakis, et al.. Hardware/compiler co-development for an embedded media processor. Proc. of IEEE, Vol. 89, Nov. 2001, pp. 1694 – 1709.

[2] M. Adiletta, et al.. The next generation of Intel IXP network processors. Intel Technology Journal, Vol. 6, Issue 3, on-line version.

[3] RAMBUS Inc.. 512/576 Mb 1066 MHz RDRAM datasheet (4i Independent Bank Architecture). [online]. Available: http://www.rambus.com/downloads/RDRAM1066.512i.0117-030.pdf.

[4] R. Crisp. Direct RAMbus technology: the new main memory standard. IEEE Micro, Vol. 17, No. 6, Nov.-Dec. 1997, pp. 18 – 28.

[5] B. Miller et al.. Two high-bandwidth memory bus structures. IEEE Design and Test of Computers, Jan.-Mar., 1999, pp. 42 – 52.

[6] RAMBUS Inc.. RDRAM for small, high performance memory systems. [online]. Available: http://www.rambus.com/downloads/Value_Proposition_Networking.pdf.

[7] J. Cong, Z. (D.) Pan. Interconnect performance estimation models for design planning. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 20, No. 6, June 2001, pp. 739 – 752.

[8] IC Knowledge. DRAM Trends. [online]. Available:http://www.icknowledge.com/trends/dram.html.

[9] J. Stokes. Ars Technica RAM Guide, Part Ⅲ: DDR DRAM and RAMBUS. [online]. Available: http://www.arstechnica.com/paedia/r/ram_guide/ram_guide.part3-1.html.

[10] K. Itoh. VLSI memory chip design. Springer 2001, Ch. 3.

[11] T. P. Haraszti. CMOS memory circuits. Kluwer Academic Publishers, pp.73 – 76.

[12] H. Schmit, et al.. PipeRench: a virtualized programmable datapath in 0.18 μm technology. In: Proc. Custom Integrated Circuits Conf., 2002, pp. 63 – 66.

[13] N. Weste and K. Eshraghian. Principles of CMOS VLSI design. Addison-Wesley, 1993, Ch. 2.

[14] Cadence. [online]. Available: http: //www.cadence.com.

[15] T. Mimura, et al.. System module: a new chip-to-chip module technology. In: Proc. Custom Integrated Circuits Conf., 1997, pp. 439 – 442.

[16] D. Burger, T. M. Austin. The simplescalar tool set version 2.0. Technical Report 1342, CS Department, University of Wisconsin, June 1997.

[17] The Standard Performance Evaluation Corporation. SPEC CPU2000 V1.2. [online]. Available: http://www.spec.org/cpu2000/.

[18] N. L. Binkert, L. R. Hsu, A. G. Saidi, R. G. Dreslinski, A. L. Schultz, S. K. Reinhardt. Performance analysis of system overheads in TCP/IP workloads. Parallel Architectures and Compilation Techniques 2005.

[19] Geek.com. HP-Compaq/DEC—Alpha 21364 (EV7, EV79) Processor Table. [online]. Available: http://ww.geek.com/procspec/dec/21364.htm.

[20] IXBT.com. Two methods for measuring memory latency on Intel Pentium 4 platform in RightMark memory analyzer—how to choose the right one? http://www.digit-life.com/articles2/cpu/rmma-p4-latency.html.

[21] J. Hennessy, D. Patterson. Computer architecture: a quantitative approach. Morgan Kaufmann, San Francisco, CA, third edition, 2002, Ch. 5.

[22] R. E. Matick, S. E. Schuster. Logic-based eDRAM: origins and rationale for use. IBM J. Research & Development, Vol. 49, No. 1, Jan. 2005, pp. 145 – 165.

[23] C. Kim, D. Burger, S. W. Keckler. Nonuniform cache architectures for wire-delay dominated on-chip caches. IEEE Micro, Vol. 23, Nov.-Dec. 2003, pp. 99 – 107.

[24] The Standard Performance Evaluation Corporation. SPEC CPU95. [online]. Available: http://www.spec.org/benchmarks.html.

[25] R. E. Matick, T. J. Heller, M. Ignatowski. Analytical analysis of finite cache penalty and cycles per instruction of a multiprocessor memory hierarchy using miss rates and queuing theory. IBM J. Research & Development, Vol. 45, No. 6, Nov. 2001, pp. 819 – 842.

[26] D. Buss, et al.. SOC CMOS technology for personal Internet products. IEEE Trans. On Electronics Devices, Vol. 50, No. 3, March 2003, pp. 546 – 556.

4 An Automatic 2.5-D Layout Design Flow

Yangdong (Steve) Deng

Institute of Microelectronics, Tsinghua University
Beijing 100084, P. R. China, dengyd@tsinghua.edu.cn

Wojciech P. Maly

Electrical and Computer Engineering Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891 USA, maly@ece.cmu.edu

Abstract The highly complex design process for 2.5-D VLSI requires automatic Electronic Design Automation tools. In this chapter we propose a layout design framework for 2.5-D ICs. The building blocks include floorplanning, placement, and routing tools, which are capable of packing a logic circuit into a stacked space.

Keywords 2.5-D integration, floorplanning, placement, routing, layout.

In the previous chapter, we introduced our design case studies with system partition performed by hand. In fact, the 4 designs we discussed are special cases in the sense that they possess either a regular structure or a clearly defined interface between

different system components. As a result, it's relatively easy to find an optimized layout implementation in a purely hand-crafted manner. For ASIC applications, however, it is usually infeasible to manually pack them in a 2.5-D layout space because of the complexity and lack of regularity. Without the help of automatic tools, the architects' choice of the system partitioning and the corresponding assignment of inter-chip contacts could be sub-optimized and inefficient in the utilization of the 2.5-D interconnect resource. For instance, when dealing with a random logic based designs with hundreds of thousands of or even millions of cells, it is difficult for a designer to make educated design decisions on how to assign the inter-chip contacts a subset of nets so that the wire length/timing can be optimized.

In this and the two succeeding chapters, we extend our feasibility study to ASIC designs by developing 2.5-D physical design tools. There are already several research projects focusing on building layout tools under the context of 3-D integration (e.g.,[1]) and 3-D System-in-Package (e.g.,[2]). In this work, we developed floorplanning, placement and routing tools for the 2.5-D integration style. Our tools provide an automatic, self-contained layout design framework for 2.5-D integrated ASICs with different design styles. In the remaining of this chapter, we will outline our 2.5-D ASIC layout design framework.

4.1 A 2.5-D Layout Design Framework

The 2.5-D layout design framework developed in this work is illustrated in Fig. 4.1. We apply different design flows for two major ASIC design styles, hierarchical and flattened styles.

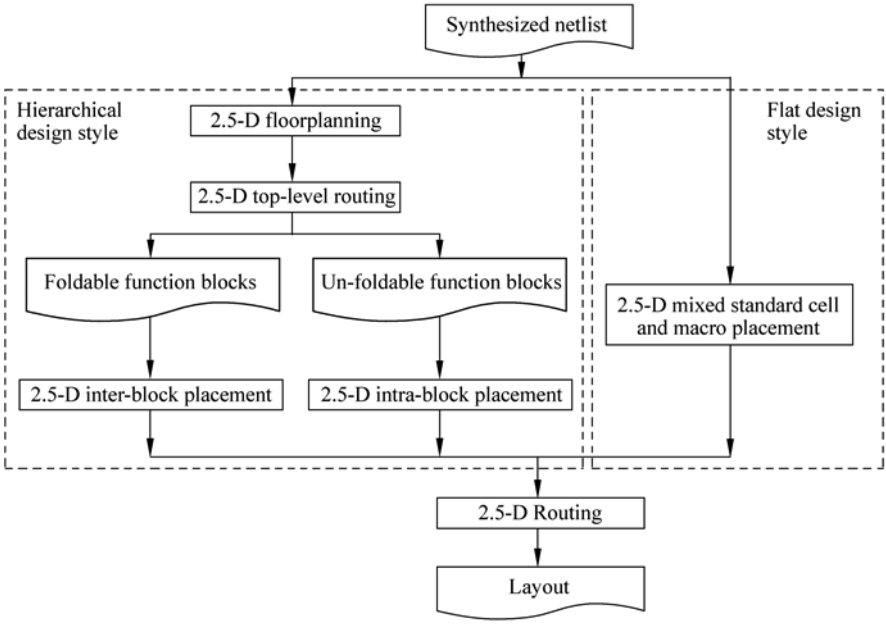


Figure 4.1 A 2.5-D layout synthesis framework

According to the hierarchical design style, the input netlist is organized as a set of interconnecting functional blocks with arbitrary rectilinear shapes and aspect ratios. A 2.5-D floorplanning tool is aimed to map each block to a unique position on a specific layer in a multi-layer chip stack. A global routing process can then be followed to assign pins on the boundary or top of the functional blocks. For the 2.5-D floorplan design problems, it is necessary to further distinguish two scenarios, designs with and without foldable blocks. A foldable block is one that can be split into multiple chips in a 2.5-D system. Typical function blocks in this category include random logic based blocks, large multi-bank memory macros, and configurable fabrics. On the other hand, analog blocks requiring careful parameter matching and highly optimized customization could only be assigned to a specific layer of chip, although the whole netlist can be partitioned into multiple chips in a

2.5-D system. No matter foldable blocks exist or not, different blocks could be placed and routed in parallel after the 2.5-D floorplan and top-level pin-assignment processes. If a foldable block is assigned to two or more layers of chips, an intra-block 2.5-D placement need to be performed so that cells can be assigned to unique positions on a certain chip. For non-foldable blocks and foldable blocks assigned to a single layer of chip, we only need to perform monolithic intra-block placement, after which, a 2.5-D routing step is followed to complete signal and power connections.

In a flattened design style, layout designers directly carry out the 2.5-D placement and routing tasks on a flatten netlist consisting of both standard cells and macros. Such a flow could usually accomplish superior solution quality, but at the cost of a longer turnaround time because of the inability to implement different blocks in parallel.

In this research, we developed 2.5-D floorplanning, placement, and routing tools, which will be briefly introduced in the remaining sections. In Chapters 5 and 6, the details of our 2.5-D floorplanning and placement tools as well as the corresponding design case studies using the tools will be further covered in depth.

4.1.1 2.5-D Floorplanning

Our 2.5-D floorplanning tool is based on a multi-layer Bounded Slice-Line (BSG) data structure[3]. Basically, a BSG data structure is maintained for each layer of chip in a 2.5-D system and the legality of a floorplan solution inside each layer could be automatically guaranteed. The optimization is performed by a highly

optimized simulated annealing engine. During the annealing process, we allow blocks switched to a different layer of chip. Our tools could also handle designs with foldable blocks. For designs in which some blocks could be split into more than one chip, the annealing engine could randomly change the area percentage of one block assigned into a given chip. We will explain our floorplanning tools in more detail when we present our floorplan level feasibility study for 2.5-D ASIC designs in the next chapter.

4.1.2 2.5-D Placement

The 2.5-D placement problem could be formulated in two different flavors: 1) pure standard cell placement in a hierarchical design style; and 2) mixed macro and standard cell placement in a flattened design style. Our 2.5-D placement tools were extended from a bi-partition placement framework, UCLA's Capo placer[4], with new heuristics to handle designs in both flavors under the 2.5-D integration paradigm. An important feature is that our tools always try to match the demand and supply for inter-chip interconnection resource during the placement process so that the inter-chip contacts can be exploited in an educated manner. We are going to explain our placement tool in more detail when we present our placement level feasibility study for 2.5-D ASIC designs in Chapter 6.

4.1.3 2.5-D Global Routing

In our 2.5-D layout design framework, we constructed a global router that could handle both monolithic and 2.5-D design implementations. After the floorplan design

and placement processes, a global routing process can be executed to complete the signal connections. Accordingly, we are able to report global-routed wire length to evaluate the quality of a design. This constructive wire length evaluation is much more accurate than the conventional bounding-box based estimation.

The VLSI routing problem is to find a connecting conductor (in terms of metal, poly, via or contact) path for every net in a circuit. Today the target layout is usually modeled as a checker board graph or routing graph as shown in Fig. 4.2. To build such a routing graph, the entire layout region is regularly divided into tiles and the size of a tile determines the granularity of abstraction (The finer is the granularity, the more accuracy is available, and vice versa.). Each tile is represented as a vertex in the routing graph. All net terminals located inside a tile are assigned to the corresponding vertex. Since a modern VLSI chip may 8 or more metal layers for routing signal and power wires, the routing graph has as many layers as the number of metal layers provided by the targeted process. Generally speaking, each layer has a preferred routing direction. Accordingly, if

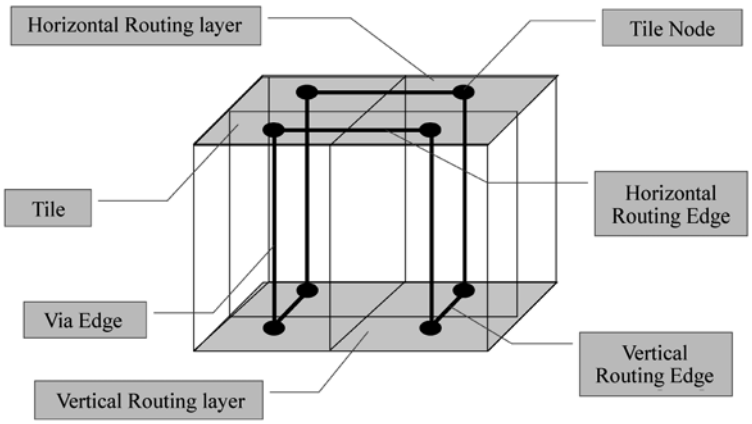


Figure 4.2 2.5-D routing graph

two vertices are horizontally (vertically) adjacent in a horizontal (vertical) routing layer, an edge exists between them. In addition, there is an edge between two tiles in neighboring routing layers. Each edge is associated with a capacity which indicates how many routes can be made through the edge.

In a 2.5-D layout, logic cells or macros could be placed on multiple device layers. From a routing point of view, however, the 2.5-D routing problem is almost identical to the monolithic routing problem. In fact, in today's typical monolithic designs, pins on a net could already be positioned on multiple metal layers.

Our global router is based on the algorithm proposed in[5]. Before performing global routing, every signal net is decomposed into a set of edges using a minimum spanning tree algorithm. Then the edges are sequentially routed by a classical maze routing engine. To avoid routing congestion, the cost of routing through a certain edge is modeled as a two-tier function with linear transition between two values. As illustrated in Fig. 4.3, if the number of routes through an edge is below a threshold value *Th* 1, the routing cost is a constant low. When the number routing is beyond another threshold value *Th* 2, the routing cost is high. For the number of routes is in between, the cost is a linear function of it. Then we perform iterative rip-up and re-route on every net to improve solution quality.

The target process is the ST Microelectronics' 0.18 μm, 6-metal CMOS process. The tile size is selected so as to contain $10-15$ standard cells with 15% empty space. In the targeted process, it is assumed that inter-chip contact has an area pitch of $5\ \mu m \times 5\ \mu m$ and a height of around $10-20$ μm. Currently we focus on stacking of 2 chips, which are face-to-face bonded. We assume the two chips in a stack have equal areas, although it's not necessary in the future. The equivalent height of inter-chip contact is around 10 μm (face-to-face bonding).

4.2 Observations

In this chapter, we proposed a complete layout design framework for the 2.5-D integration scheme. This framework consists of layout synthesis tools developed to fit the requirements of different design styles at major VLSI physical design stages.

With these tools, we could then compare the interconnection characteristics between the monolithic and 2.5-D implementations of a given design. Both total wire length and worst-case wire length are assessed for this purpose. The former generally reflects the routability of a design, while the worst-case wire length tends to have a significant impact on the system performance in today's SoC designs. For instance, the Cell processor for PlayStation 3 has ~2000 wires longer than 1 cm and these wires have been the determining factor on the timing performance[6]. Consequently, we would conclude that 2.5-D implementation outperforms its monolithic equivalent if a better wire length distribution is observed. Of course, the wire length comparison has its limitations because the worst timing path may not coincide with the longest timing path. One important extension to this work will be to develop a timing analysis mechanism for the 2.5-D designs based on the routing information provided by our router.

The feasibility investigation is conducted at two major abstraction levels during physical design: floorplan level (functional blocks) and placement (standard cells and macros) level. In the succeeding two chapters, we will discuss the feasibility at floorplan and placement levels, respectively.

References

[1] S. Das. Design automation and analysis of three-dimensional integrated circuits. Ph. D. dissertation, Department of Electrical and Computer Science, MIT, May 14, 2004.

[2] S. K. Lim. Physical Design for 3D System-On-Package: Challenges and Opportunities. IEEE Design & Test of Computers, Vol. 22, No. 6, 2005, pp. 532 – 539.

[3] S. Nakatake, et al.. Module placement on BSG-structure and IC layout applications. In: Proc. of Int'l Conf. Computer Aided Design, 1996, pp. 484 – 491.

[4] A. E. Caldwell, A. B. Kahng, I. L. Markov. Can recursive bisection alone produce routable placements? In: Proc. Design Automation Conf., 2000, pp. 477 – 482.

[5] J. Cong, P. Madden. Performance driven multi-layer general area routing for PCB/MCM designs. In: Proc. Design Automation Conf., June 1998, pp. 356 – 361.

[6] S. R. Mraz. Dissecting the PS3 'Cell' architecture. PS3Insider, [online]. Available: http://www.ps3insider.com/modules.php?name = Content&pa = showpage&pid = 3.

5 Floorplanning for 2.5-D Integration

Yangdong (Steve) Deng

Institute of Microelectronics, Tsinghua University
Beijing 100084, P. R. China, dengyd@tsinghua.edu.cn

Wojciech P. Maly

Electrical and Computer Engineering Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891 USA, maly@ece.cmu.edu

Abstract We introduce the floorplanning techniques for the 2.5-D integrated VLSI systems in this chapter. Three different 2.5-D/3-D floorplanning problems are established targeting major layout design styles. The solution techniques are proposed accordingly. The results show that the 2.5-D layout solution considerably reduces wire length of interconnects and thus delivers a higher system performance. We also propose thermal driven floorplanning techniques. By taking into account temperature profile during the floorplanning optimization process, it's possible to effectively control the worst-case temperature on the chip at a small penalty of interconnect length.

Keywords 2.5-D integration, floorplanning, bounded-slice line, simulated annealing, temperature distribution, thermal-driven.

In the previous chapter, we briefly outlined our 2.5-D layout synthesis framework. In this chapter, the 2.5-D layout tools are applied on floorplan level designs. We will introduce our floorplanning algorithms along with the design case studies in this chapter.

Floorplan design is the starting point of today's physical flows. Under the monolithic integration setup, the input to the floorplan design problem is a set of function blocks that have a rectilinear shape. The objective is to map each functional block in a design to a unique position on the layout so that certain design objectives could be optimized. Commonly used objectives include wire length, critical path delay, and chip area. Recently, new objectives like thermal distribution and design for manufacturability might also need to be taken into account. A floorplan solution is valid only when there is no overlap between any two blocks. For digital designs, key physical design steps such as placement, global interconnection planning, buffer insertion, top-level pin assignment, and power/ground routing can be then performed on the basis of floorplan solutions. In today's typical ASIC design flows (e.g., [1,2]), floorplan design is often performed before detailed RTL synthesis so that valuable physical information can be derived and then feed to the synthesis engine. For analog designs, which typically designed in to bottom-up manner, a floorplan design followed by a full-chip routing would complete the layout design. Floorplan design has attracted a large body of research work. Interested readers please refer to References[3−9] for more details.

Under the 2.5/3-D integration context, the floorplanning problem can be formulated with different flavors according to the architecture of the designed system. Here we can classify VLSI system into three categories:

Category 1: random logic dominated architecture

Typical applications of this category include graphic processing unit (GPU) like

NVidia's G80 and network processor like Motorola's C-Port. For instance, NVidia[10] reports that 78% area is devoted to random logic and 20% area is for small memory blocks embedded into logic on the die of NV30 GPU. Obviously, in a 2.5/3-D implementation, almost all the functional blocks can be "folded" into multiple layers. Accordingly, the 3-D floorplan can be simply formed by migrating a monolithic floorplan. For instance, after a monolithic floorplan is constructed, we can decompose each block into two halves so that each half has a width (height) corresponding to 0.707 of the original width (height). Thus the total wire length of top-level nets will be scaled down by 0.707. For applications of the first category, it is readily to see that the floorplanning problem can be solved using a monolithic floorplanning algorithm. We will not consider this category in the remaining parts of this paper.

Category 2: Custom design dominated architecture

General-purpose microprocessors and RF transceivers are typical applications of this category. In the 3-D implementation, large custom blocks have to reside on a specific device layer. For example, the inner core execution logic of a microprocessor is aggressively (usually manually) optimized with regard to the fabrication process and it's generally beneficial to keep its circuitry in one layer of chip. In addition, analog modules such as an inductance ring and oscillator are not allowed to be divided into two parts and then assigned to different layers. For this category, 3-D floorplanning is a multi-layer placement problem where each module can be allocated to one and only one layer. For custom design dominated designs belonging to Category 2, the 3-D floorplanning problem is a multi-layer extension of the traditional floorplanning problem a multi-layer assignment problem:

Given a set of modules M: {M_i} and a set of nets N: {N_i} interconnecting those modules in M, find a legal (no overlapping) embedding of them in a layered

space (where each module can be allocated to one and only one layer) so that total chip area, total wire length, or a weighted sum of them, can be minimized.

This formulation also applies to the situations where the footprint of inter-chip contacts is relatively big, e.g., 50 μm × 50 μm (i.e. similar to that of flip-chip interconnection). Here inter-chip contacts should be considered as a coarse-grained resource and be assigned during the floorplanning stage.

Category 3: Intermediate architecture

This case is actually a generalization of the above two categories. In the systems belonging to this category, some functional blocks can span multiple layers of chips, while others can only reside in one layer. As a result, a floorplanning engine has to be able to optimize the location of these two kinds of blocks. Accordingly, the 3-D floorplanning problem can be formulated as:

Given a set of modules M: {M_i} among which a subset MF: {MF_i} can simultaneously reside on multiple layers, and a set of nets N: {N_i} interconnecting those modules in M, find a legal (no overlapping) embedding of them in a layered space so that total chip area, total wire length, or a weighted sum of them, can be minimized.

Besides traditional optimization objectives, a 3-D VLSI system implies a larger power density than its monolithic equivalent does. On-chip hot spots could incur serious degradations of system performance, power consumption, and reliability. However, we do observe that a 2.5/3-D floorplan without thermal constraints could lead to a maximum temperature of 180℃ and a temperature gradient of 152℃[11] with traditional air-cooling techniques. Accordingly, it's of key importance to avoid excessive heat build-up and temperature difference in a 3-D integrated system. In other words, the thermal objective must be taken into account in the 3-D floorplanning problems formulated above.

5.1 Floorplan Level Evaluation—Category 2 Circuits

Let's first consider the 2.5-D floorplanning problem on Category 2 circuits. The problem is a simultaneous partition and assignment problem: the input netlist has to be partitioned into multiple parts with each part assigned to a different chip in a 2.5-D system, and within the given chip every function block has to be placed without overlapping with other blocks.

5.1.1 Technique

Our floorplanning algorithm is based on the Bounded Slice-line Grid (BSG) structure proposed by Nakatake et al. in[12]. As a compact representation for non-slicing floorplan, the BSG data structure encodes a large solution space including the optimal one and allows rapid exploration. To perform 2.5-D floorplanning, we maintain a multi-level BSG data structure, with one BSG for each layer of chip in the stack. The data structure is illustrated in Fig. 5.1. Obviously, the multi-level BSG provides a complete solution space for the 2.5-D floorplanning problem. We also developed a traditional 2-D floorplanner, which is a straightforward implementation of the algorithm described by Nakatake et al. in [12].

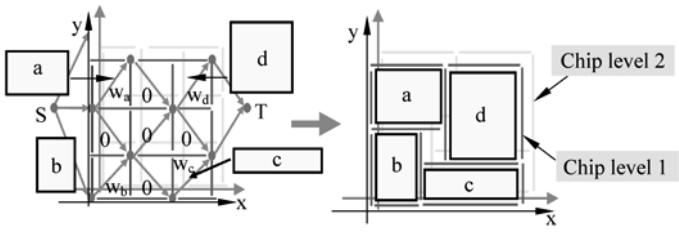


Figure 5.1 2.5-D floorplanning

The optimization is accomplished through a simulated annealing engine with cost function given by Equation (5.1). The cost function is the weighted sum of three components: total wire length, floorplan area, and total number of inter-chip contacts.

$$\text{Cost} = \lambda \cdot \text{wirelength} + \gamma \cdot \text{chip\_area} + \mu \cdot \text{num\_2.5D\_vias} \qquad (5.1)$$

In our implementation, we choose λ and γ so that the first two terms roughly equal. Thus the floorplanner puts almost equal efforts to optimize wire length and chip area. The third term is used to indirectly control the number of inter-chip contact. The value of m is used to represent "cost" of inter-chip contact.

During simulated annealing process, neighboring solutions are explored by random perturbations on the current solution. In our implementation, a random perturbation, or a random move, can be one of one the three following ways: displacing a block to another position, changing a block's aspect ratio, and swapping two blocks' positions. A block can be removed from the BSG of one level and inserted into the BSG of another level.

The cooling schedule is set as follows:

1. Starting temperature According to Huang et al.[13], we set the starting temperature as 20 times of the standard deviation of a certain number of random perturbations. The idea is to keep the accept ratio close to 1 at the initial stage of simulated annealing.

2. Temperature updating After a given number of random moves (we set number of inner loops as 100 times of the number of movable objects), current temperature is updated by multiplied by a factor k. The value of k is calculated as a function of accept ratio a. We set $k = 0.6$ when $0.95 \leqslant a < 1$; $k = 0.9$ when $0.6 \leqslant a < 0.95$; $k = 0.99$ when $0.05 \leqslant a < 0.6$; $k = 0.8$ when $a \leqslant 0.05$.

3. Freezing temperature The simulated annealing process is terminated when the cost variation is less than 1/1000 for three consecutive temperatures.

5.1.2 Results

In our experiments, we use the largest three MCNC benchmark circuits, ami33, ami49 and playout[14]. Details of these circuits are shown in Table 5.1. Both 2D and 2.5-D floorplans of ami49 are shown in Fig. 5.2. The 2.5-D floorplan consists of two monolithic floorplans, which would be face-to-face bonded. The wire length reduction for a given net is clearly illustrated. In the monolithic floorplan of Fig. 5.2(a), net 31 spans a long route connecting three modules. With the flexibility provided by an inter-chip contact, its length can be greatly reduced in the 2.5-D floorplan.

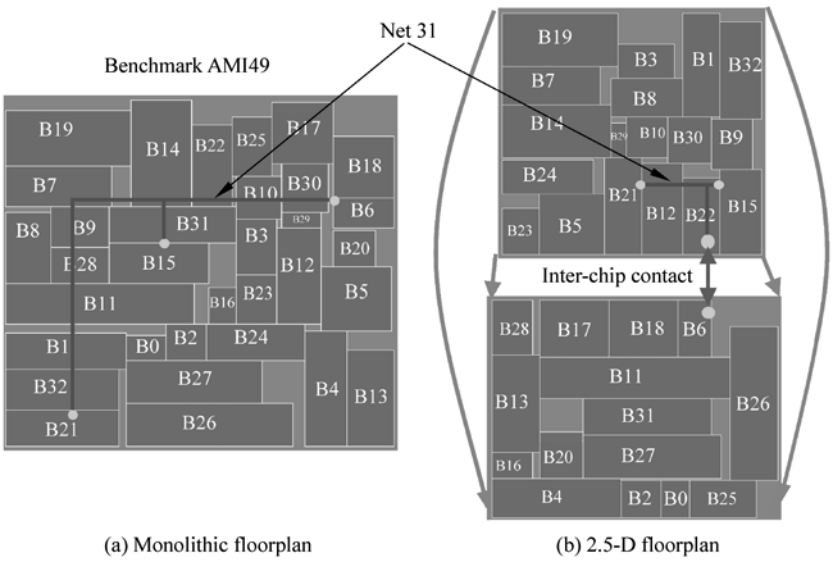


Figure 5.2 A floorplan example

Table 5.1 compares the monolithic and 2.5-D floorplanning solutions in terms of three design metrics: total chip area, total wire length, and worst-case wire length. The latter one, worst-case wire length, is computed as the total length of the 10 longest nets. Because we assume an over-the-cell routing model, the area reductions in 2.5-D implementation are negligible. However, in the 2.5-D floorplans of the above three benchmark designs, we observe more than 30% reductions in the total wire length. Meanwhile, the worst-case wire lengths in 2.5-D floorplan are up to 33% shorter. The wires handled at this stage are all global wires, which are very likely to appear in the critical path and thus have a significant impact on the system delay. As a result, the wire length reductions imply potential for significant performance improvement. It also implies that the 2.5-D stacked implementation needs fewer buffers, which could be power hungry and hard to place.

Table 5.1 2-D and 2.5-D floorplans for Category 2 designs

		2-D Floorplan	2.5-D Floorplan	Reduction
ami33 (33 Modules, 123 Nets)	Total Area	1316140	1232938	6%
	Worst-Case Wirelength	2923	2457	16%
	Total Wirelength	81351	57314	30%
ami49 (49 Modules, 408 Nets)	Total Area	44096472	43200556	2%
	Worst-Case Wirelength	12005	8099	33%
	Total Wirelength	894100	625769	30%
playout (62 Modules, 2506 Nets)	Total Area	120797642	116981524	3%
	Worst-Case Wirelength	18299	14054	23%
	Total Wirelength	5166374	3163054	39%

5.2 Floorplan Level Evaluation—Category 3 Circuits

Typical System-on-Chip and ASIC designs would contain a large number of random logic based blocks. If a functional block is mainly composed of standard cells, it is actually to split it into two or more chips in a 2.5-D system, as long as the inter-chip contacts could provide proper connections between adjacent layers. In other words, our previous formulation of the 2.5-D floorplanning problem actually over-constrains the solution space. Accordingly, in this section we consider the 2.5-D floorplanning problem for Category 3 circuits.

5.2.1 Technique

Generally not every functional block would only reside on one specific layer of chip. To handle the foldable blocks introduced in the last chapter, we introduce a shadow block for each of the potentially multi-level blocks (This discussion is for stacking of two layers of chips, but can be generalized to the case of stacking more than two layers of chips). It is assumed that the area of such a block can flow from it to its shadow block and *vice versa*. Of course, a potentially multi-level block can also be completely placed in one layer, and if this is the case its shadow block would have an area of 0. The idea of shadow blocks is illustrated in Fig. 5.3.

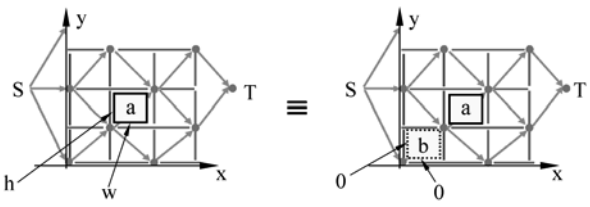


Figure 5.3 Insert a 0-weight cell

A related problem is: should a module and its shadow block be exactly aligned (say both have exactly the same left bottom position)? There are some automatic solutions to align two functional blocks: 1) one solution is to introduce artificial nets with large weights between them, and 2) use extra edges in the constraint graph[15−17]. However, when using Solution 1 a lot of optimization effort will be spent on the artificial nets, while Solution 2 will change the solution structure. As a matter of fact, we believe that positions of a potentially multi-level block and its shadow block should be naturally determined in the optimization process and no hard alignment constraints would be necessary.

With the concept of shadow block, we are able to extend our floorplanning algorithm to handle the second formulation of 2.5-D floorplanning problem.

5.2.2 Results

Under the formulation of 2.5-D floorplanning problem, again we generate 2.5-D floorplans for the largest 3 MCNC benchmarks. For each benchmark circuit, we randomly pick 1/3 of the blocks and allow them to be split. Table 5.2 compares total chip area, total wire length and longest wire length between monolithic. The area and wire length reductions are comparable with the first formulation. With the extra flexibility, we could observer better wire length results than we did with the first formulation. Another key advantage is that this formulation is compatible with a hierarchical layout design flow and the floorplanning solution can serve as the starting point for the succeeding placement procedure.

Table 5.2 2-D and 2.5-D floorplans for Category 3 designs

		2-D Floorplan	2.5-D Floorplan	Reduction
ami33 (33 Modules, 123 Nets)	Total Area	1316140	1262289	4.1%
	Worst-Case Wirelength	2923	2822	3.5%
	Total Wirelength	81351	55314	32.0%
ami49 (49 Modules, 408 Nets)	Total Area	44096472	37170640	15.7%
	Worst-Case Wirelength	12005	7350	38.8%
	Total Wirelength	894100	521244	41.7%
playout (62 Modules, 2506 Nets)	Total Area	120797642	110529842	8.5%
	Worst-Case Wirelength	18299	11968	34.6%
	Total Wirelength	5166374	3156654	38.9%

5.3 Thermal driven floorplanning

Compared to the conventional monolithic (2-D) integration, one leading concern in designing 2.5-D systems is the heat dissipation. This is an important factor to consider because of the following reasons. It is true that vertically stacking multiple chips will help reduce the length of global wiring and reduce the power consumption associated with global interconnects. However, 2.5-D integration will be most likely applied to high performance designs in which a significant amount power will still be dissipated even with certain reduction in the global interconnect power. The high power density in these applications already brings challenges in cooling traditional monolithic chip designs and will certainly exacerbate heat removal in 2.5-D/3-D chips. The latter is primarily due to the

increase of thermal resistances along the major heat dissipation paths in the vertical direction caused by multiple stacked chip layers. The elevated chip temperature will also introduce long term reliability issues. Furthermore, timing failures may be resulted if a significant temperature gradient exists laterally or across different chip layers. Another important concern for modern VLSI technologies is the increasing leakage power that must be considered in design optimization[18]. As a matter of fact, leakage power is strongly dependent on chip temperature and must be analyzed together with temperature. The strong interdependency between temperature and leakage power is due to exponential dependency of leakage on temperature as well as the positive feedback between the two.

In this section, we propose a temperature-aware 3-D IC floorplanning methodology considering the interaction between temperature and leakage power. We address the optimization of 2.5-D designs at the floorplan level by simultaneously considering chip area, wire length, chip temperature and gradient as well the temperature-dependent leakage power. Although the proposed technique can be applied to general 3-D chip designs, we focus only on the special case where only two chip layers are stacked vertically given the immediate technological feasibility of such a choice. As shown in Fig. 5.4, to enable a feasible temperature-driven floorplanning methodology, we start by performing full-chip thermal and leakage pre-characterization based on detailed thermal models. As such, the temperature and its gradient of both device layers can be efficiently evaluated in a way such that the important temperature dependent leakage power is brought into the consideration. Extended from a BSG-based[12] floorplanning algorithm, our floorplanner can efficiently optimize total chip area, wire length, maximum chip temperature and temperature gradient simultaneously. Our experiments have shown that the proposed floorplanning technique can significantly reduce the

maximum temperature and the gradient on the two chip layers while introducing only a mild overhead in the chip area and the total wire length. Furthermore, our floorplanner is capable of providing a continuous tradeoff between temperature and performance (measured by wire length) in a way that the 3-D floorplanning engine can be tuned for specific design needs.

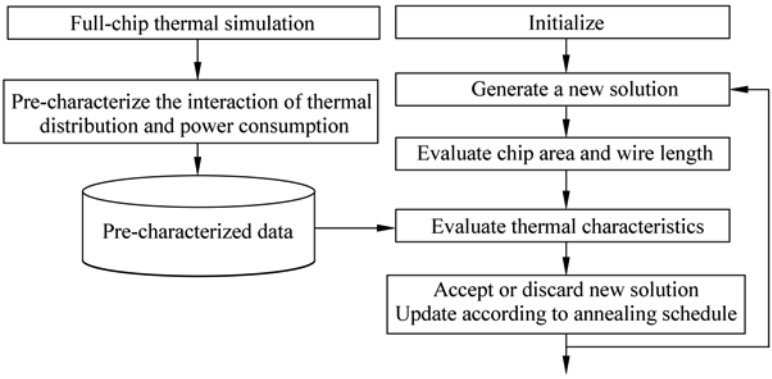


Figure 5.4 2.5-D thermal-driven floorplanning flow

In the past, thermal and leakage issues have attracted significant interests in research community that include full-chip thermal analysis and cell placement for the conventional 2-D chips[19–24], architecture level thermal/leakage management and optimization[18,25,26] and very recently physical design for 3-D chips (e.g., [27–29]). To our best knowledge, the work reported in this paper is the first one to perform 3-D floorplanning with detailed temperature and leakage information and their interdependency derived from a full-chip thermal simulator.

5.3.1 Chip Level Thermal Modeling and Analysis for 2.5-D Floorplanning

A two chip-layer 3-D IC with a flip chip type package is shown in Fig. 5.5. As

can be seen in the figure, the primary heat removal paths are from the packaged chip layers to the heat sink at the top and to the PCB board at the bottom. Since the side walls of the chip are much smaller compared to the lateral dimensions, we treat them as thermally reflective, i.e. heat flow through the side walls is neglected. The boundary conditions at the PCB board and the ambient (through the heat sink) are modeled as convective. The thermal impacts of various packing interface material layers and interface materials are modeled using 1-D dimensional equivalent thermal resistances with proper values[19,23].

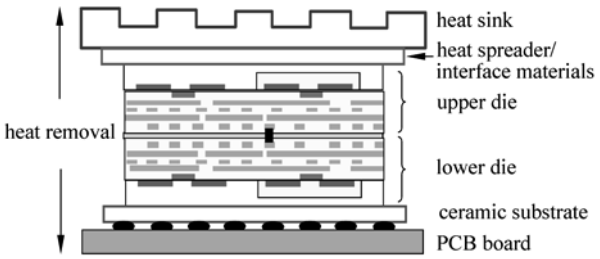


Figure 5.5 A 3-D IC with two stacked chip layers in a package

For the two vertically stacked dies, significant thermal gradients may exist due to non-uniform on-chip power distributions. To capture such non-uniform on-chip profiles, detailed thermal modeling and analysis is employed to achieve good accuracy. For this purpose, we adopt an efficient full-chip thermal simulator[23] that is based on detailed finite difference discretization of the following governing heat transfer partial differential equation

$$\rho c_p \frac{\partial}{\partial t} T(\vec{r},t) = \nabla \cdot (k(\vec{r},T)\nabla T(\vec{r},t)) + g(\vec{r},t) \tag{5.2}$$

subject to the general boundary condition

$$k(\vec{r},T)\frac{\partial}{\partial n_i}T(\vec{r},t)+h_iT(\vec{r},t)=f_i(\vec{r},t) \tag{5.3}$$

where T is the temperature, $\vec{r}$ denotes the location in 3-D, ρ is the material density, c_p and k are the specific heat of the thermal conductivity of the material, g is the power density of the heat sources, n_i is the outward direction normal to the boundary surface, h_i is the heat transfer coefficient, and f_i is an arbitrary function at the boundary surfaces.

To characterize the relationship between the power sources and the generated on-chip temperature profile, in the first step of our thermal/leakage pre-characterization, the power to heat mapping within and between active device layers is calculated. In this work, it is assumed that the amount of heat generated by self-heating of interconnects can be neglected and only the power dissipation due to the transistor switching activities is considered. Hence, the power is only assumed to be generated at the transistor layers that are within a thin depth from the silicon substrate for each stacked die.

To consider the temperature dependent leakage power consumption, it is desired to determine the transistor layer temperature distribution due to the power dissipated. For this purpose, the two transistor layers are laterally partitioned into $M=2\times N\times N$ bins based on a user-specified granularity. Then, full-chip thermal simulation is applied to compute the average temperature increase in all bins if a unit power is dissipated in any of these bins. The power and temperature interactions for a particular power dissipating bin are illustrated in Fig. 5.6. A total of $M\times M$ interactions will be extracted at this stage such that for any given on-chip power dissipation distribution, the temperature increase at any of the two transistor layers can be determined by superposition at the specified granularity[23]. Stated mathematically, we have

$$\Delta \boldsymbol{t} = \boldsymbol{F}_{p2t} \cdot \boldsymbol{p} \tag{5.4}$$

where $\boldsymbol{p} = [p_1, p_2, \cdots, p_M]^{\mathrm{T}}$ is the power dissipations in M bins, $\Delta \boldsymbol{t} = [\Delta t_1, \Delta t_2, \cdots, \Delta t_M]^{\mathrm{T}}$ is the temperature increases (with respect to ambient temperature) due to $\boldsymbol{p}$ in these bins, and $\boldsymbol{F}_{p2t}$ is a $M \times M$ matrix representing the mapping from $\boldsymbol{p}$ to $\Delta \boldsymbol{t}$

$$\boldsymbol{F}_{p2t} = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,M} \\ f_{2,1} & f_{2,2} & \ddots & f_{2,M} \\ \vdots & \cdots & \ddots & \vdots \\ f_{M,1} & f_{M,2} & \cdots & f_{M,M} \end{bmatrix} \tag{5.5}$$

where $f_{i,j}$ relates the power source applied in bin j to the temperature increase observed in bin i. It should be noted, however, a fine discretization step smaller than the thermal characterization granularity can be chosen in order to ensure the accuracy of the average temperature increase computation for each bin.

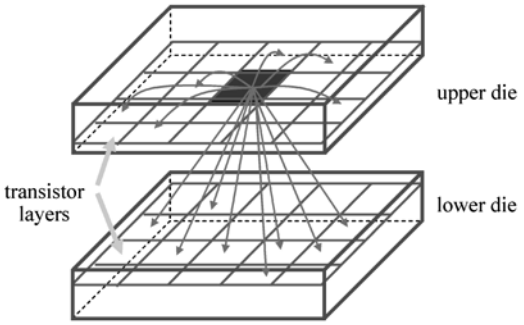


Figure 5.6 Thermal interactions between a region of the top transistor layer to all other regions on both transistor layers (not all interactions are drawn)

It has been shown that for a given floorplan detailed full-chip thermal analysis can be employed to pre-characterize the interactions of the power dissipation and

the temperature distribution. However, during the optimization of the 2.5-D/3-D IC floorplan, many floorplan configurations with varying total chip area and aspect ratio need to be evaluated. To avoid calling the chip-level thermal analysis for each encountered floorplan during the optimization, the following approach is adopted as shown in Fig. 5.7. A set of floorplans with different total chip area and aspect ratio are simulated and the corresponding $\boldsymbol{F}_{p2t}$ is saved in a lookup table. In the floorplanning stage, to evaluate an arbitrary floorplan, four neighboring floorplans are found from the lookup table using the area and the aspect ratio as indexes. Finally, $\boldsymbol{F}_{p2t}$ of the floorplan under evaluation is obtained by linear interpolation using the four neighboring floorplans. Our experiments show that the error introduced in this approach is not significant and can be also controlled by increasing the size of the pre-characterized lookup tables.

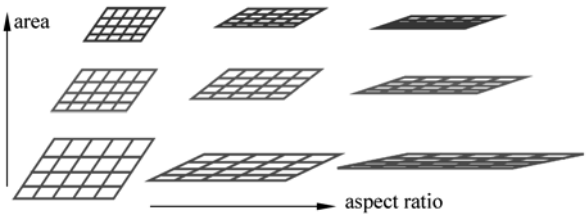


Figure 5.7 Thermal simulation of a set of floorplans with varying total area and aspect ratio (only one stacked layer is shown for each case)

5.3.2 Coupled Temperature and Leakage Estimation

In the previous section, the power to temperature mapping is pre-characterized so that the transistor layer temperature can be computed knowing the power distribution. Although the dynamic power consumption of the design may be pre-determined

or estimated using certain estimation techniques, the leakage power is strongly temperature dependent and at the same time contributes to temperature increase. Hence, the interdependency between (leakage) power and the temperature must be brought into consideration especially due to the fact the leakage may consist of a significant portion of the total power consumption.

The interdependency between leakage and temperature can be ideally considered by conducting coupled electro-thermal analysis in an iterative manner as follows[19,22]. Starting from an initial on-chip temperature distribution hence an initial leakage power distribution, an IC thermal analysis can be performed to compute the new temperature distribution using the current power consumption as the input. Then, the leakage power of all devices is modified based on the modified temperature just computed and the thermal analysis is started again using the updated total power consumption as input. The above process continues till both the temperature and leakage power distributions converge.

In the floorplanning stage, many floorplan solutions will be evaluated in terms of area, total wire length and temperature. The iterative nature and the inherent complexity of the chip-level thermal analysis, including the above coupled electro-thermal analysis at the inner loop of the optimization, will make the floorplanning extremely costly. To reduce the cost of the optimization step, a good estimation of the temperature dependent leakage power will be sought to effectively guide the proposed temperature-aware floorplanner. The final optimal floorplan, however, can be accurately evaluated by a more accurate thermal/leakage analysis, for instance, based on a piecewise linear temperature dependent leakage model.

To seek a relatively simple correspondence between the leakage and the temperature, a linear model is adopted as illustrated in Fig. 5.8[30]. For each

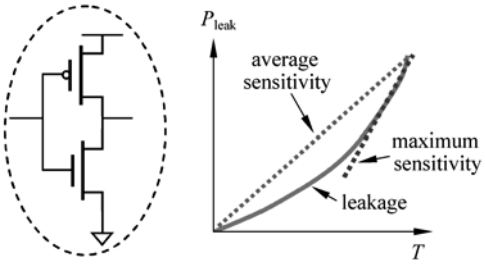


Figure 5.8 Modeling the temperature dependency of the leakage power using a linear model

active device, or more generally a partitioned circuit block, the average or the maximum temperature sensitivity of the leakage power can be selected within a certain range of temperature variation. While the former choice corresponds to an average temperature dependency the latter represents the worst-case dependency. Relating the leakage power distribution with the temperature distribution using such linear model leads to

$$\boldsymbol{p}_{\text{leak}} = \boldsymbol{p}_{\text{leak},0} + \boldsymbol{\Phi} \cdot \Delta \boldsymbol{t} \tag{5.6}$$

where with the temperature increase $\Delta \boldsymbol{t}$ (over ambient temperature), $\boldsymbol{p}_{\text{leak}} = [p_{\text{leak},1}, p_{\text{leak},2}, \cdots, p_{\text{leak},M}]^{\text{T}}$ is the leakage power of the M bins, $\boldsymbol{p}_{\text{leak}\,0} = [p_{\text{leak}\,0,1}, p_{\text{leak}\,0,2}, \cdots, p_{\text{leak}\,0,M}]^{\text{T}}$ is the leakage power at the ambient temperature, and $\boldsymbol{\Phi}$ is a diagonal matrix given by

$$\boldsymbol{\Phi} = \begin{bmatrix} \alpha_1 p_{\text{leak}\,0,1} & & & \\ & \alpha_2 p_{\text{leak}\,0,2} & & \\ & & \ddots & \\ & & & \alpha_M p_{\text{leak}\,0,M} \end{bmatrix} \tag{5.7}$$

where α_i is the temperature sensitivity of the leakage power for the devices

located in bin *i*.

Next, a relationship between the dynamic power and the temperature increase will be derived while considering the temperature dependent leakage power. Substituting (5.6) into (5.4) and expressing the total power as the sum of the dynamic power and the leakage power $\boldsymbol{p} = \boldsymbol{p}_{\mathrm{d}} + \boldsymbol{p}_{\mathrm{leak}}$ gives

$$\Delta \boldsymbol{t} = (\boldsymbol{I} - \boldsymbol{F}_{p2t}\boldsymbol{\Phi})^{-1}\boldsymbol{F}_{p2t}(p_{\mathrm{leak},0} + \boldsymbol{p}_{\mathrm{d}}) \tag{5.8}$$

Finally, using (5.6) the leakage power is given by

$$\boldsymbol{p}_{\mathrm{leak}} = (\boldsymbol{I} + \boldsymbol{\Phi}(\boldsymbol{I} - \boldsymbol{F}_{p2t}\boldsymbol{\Phi})^{-1}\boldsymbol{F}_{p2t})\boldsymbol{p}_{\mathrm{leak},0} + \boldsymbol{\Phi}(\boldsymbol{I} - \boldsymbol{F}_{p2t}\boldsymbol{\Phi})^{-1}\boldsymbol{F}_{p2t}\boldsymbol{p}_{\mathrm{d}} \tag{5.9}$$

Defining $\tilde{\boldsymbol{F}}_{p2t} = (\boldsymbol{I} - \boldsymbol{F}_{p2t}\boldsymbol{\Phi})^{-1}\boldsymbol{F}_{p2t}$, (5.8) will be simplified to

$$\Delta \boldsymbol{t} = \tilde{F}_{p2t}(\boldsymbol{p}_{\mathrm{leak},0} + \boldsymbol{p}_{\mathrm{d}}) \tag{5.10}$$

Given $\tilde{\boldsymbol{F}}_{p2t}$, the temperature increase can be rather efficiently computed via a matrix-vector multiplication. Hence, one may attempt to pre-characterize $\tilde{\boldsymbol{F}}_{p2t}$ using the lookup table based on approach presented in the previous section in order to evaluate $\Delta \boldsymbol{t}$ while avoiding expensive dense matrix factorization on the fly. To define $\tilde{\boldsymbol{F}}_{p2t}$, however, one needs to have $\boldsymbol{\Phi}$ that define the linear dependency of the leakage power on the temperature for each bin. As shown in Fig. 5.9, the leakage power will be confined to the regions where active devices are placed. Therefore, $\boldsymbol{\Phi}$ depends on the floorplan and $\tilde{\boldsymbol{F}}_{p2t}$ cannot be pre-characterized.

In this work, the package and heat sink of the 2.5-D/3-D IC are assumed to be thermally well designed such that the system is thermal-runaway free. Thermal runaway will happen if the positive feedback between the temperature and the

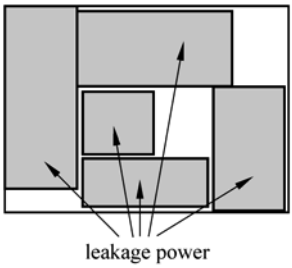


Figure 5.9 Leakage power distribution is confined within the placed circuit blocks

leakage is too strong such that a huge amount of heat is generated exceeding the heat removal capacity of the package and the heat sink. To address the difficulty described in the previous section, the special problem property inherent in those thermally safe designs will be exploited. The following theoretical results on matrix properties are first presented[31].

Proposition 1: *The series* $\sum_{i=0}^{\infty} \boldsymbol{A}^i$ *converges if and only if* $\rho(\boldsymbol{A}) < 1$. *Under the same condition,* $\boldsymbol{I} - \boldsymbol{A}$ *is nonsingular and the above series converges to* $(\boldsymbol{I} - \boldsymbol{A})^{-1}$.

In the above proposition, $\rho(\boldsymbol{A})$ is the spectral radius of matrix $\boldsymbol{A}$ and it is equal to the largest absolute value of A's eigenvalues.

Theorem 1: *For any nonnegative matrix* $\boldsymbol{A}$, $(\boldsymbol{I} - \boldsymbol{A})^{-1}$ *is nonsingular and nonnegative if and only if* $\rho(\boldsymbol{A}) < 1$.

Notice that a matrix is nonnegative if all its elements are greater or equal to zero. For our thermal/leakage problem, it is straightforward to see both $\boldsymbol{F}_{p2t}$ and $\boldsymbol{\Phi}$ are nonnegative, hence $\boldsymbol{F}_{p2t}\boldsymbol{\Phi}$ is also nonnegative. It is worth noting that $\tilde{\boldsymbol{F}}_{p2t} = (\boldsymbol{I} - \boldsymbol{F}_{p2t}\boldsymbol{\Phi})^{-1}\boldsymbol{F}_{p2t}$ relates the dynamic power consumption of the design to the temperature increase while taking into account the temperature dependent leakage power. Since the temperature increases with the power and the leakage

increases with temperature, $\tilde{\boldsymbol{F}}_{p2t}$ must be a nonnegative matrix under normal conditions. It follows from Proposition 1 and Theorem 1 that for $(\boldsymbol{I} - \boldsymbol{F}_{p2t}\boldsymbol{\Phi})$ to be nonsingular or $\tilde{\boldsymbol{F}}_{p2t}$ to be well defined, it must be true that $\rho(\boldsymbol{F}_{p2t}\boldsymbol{\Phi}) < 1$.

However, the above the result will only hold when $\tilde{\boldsymbol{F}}_{p2t}$ exists and it will not be the case if thermal runaway happens under our linear model. In the latter case, $\tilde{\boldsymbol{F}}_{p2t}$ is no longer well defined and the design will burn.

In this paper, we impose $\rho(\boldsymbol{F}_{p2t}\boldsymbol{\Phi}) < 1$ as a *thermal runaway design constraint* that must be satisfied by the package and heat sink under our linear temperature dependent leakage model. Furthermore, we assume that enough design margins exists such that $\rho(\boldsymbol{F}_{p2t}\boldsymbol{\Phi})$ will be sufficiently away from 1. Under the above conditions, $\tilde{\boldsymbol{F}}_{p2t} = (\boldsymbol{I} - \boldsymbol{F}_{p2t}\boldsymbol{\Phi})^{-1}\boldsymbol{F}_{p2t}$ can be expanded as a converging series according to Theorem 1 as

$$\tilde{\boldsymbol{F}}_{p2t} = (\boldsymbol{I} - \boldsymbol{F}_{p2t}\boldsymbol{\Phi})^{-1}\boldsymbol{F}_{p2t} = \boldsymbol{F}_{p2t} + (\boldsymbol{F}_{p2t}\boldsymbol{\Phi})\boldsymbol{F}_{p2t} + (\boldsymbol{F}_{p2t}\boldsymbol{\Phi})^2\boldsymbol{F}_{p2t} + \cdots \qquad (5.11)$$

Equally important, since the spectral radius of $\boldsymbol{F}_{p2t}\boldsymbol{\Phi}$ is sufficiently away from 1, the above series converges *fast*. This implies that truncating the series after the first few terms will lead to a good approximation of $\tilde{\boldsymbol{F}}_{p2t}$. In our implementation, we keep the first two terms in the expansion and evaluate the temperature and the leakage distributions using equations(5.8)) and (5.9) accordingly. The benefit of the approach is apparent. Instead of solving a very expensive dense matrix factorization problem for every floorplan solution, the dominant cost in our approach has been reduced to two (dense) matrix-vector products. It should be noted that this approximate approach is only used to efficiently evaluate each floorplan and guide the floorplanner. More accurate thermal verification can be performed for the final floorplan.

5.3.3 2.5-D Thermal Driven Floorplanning Techniques

We extended out 2.5-D floorplan design tool introduced in the previous sections to handle thermal effects. Initially, the 2.5-D/3-D floorplanning optimization is accomplished through a simulated annealing engine with a cost function defined as:

$$\text{Cost} = \lambda \cdot \text{wirelength} + \gamma \cdot \text{chip\_area} + \mu \cdot \text{num\_inter\_chip\_contacts} \quad (5.12)$$

The cost function is the weighted sum of three components, total wire length, layout area, and total number of inter-chip contacts (i.e. interconnects between two chips). The three components have different implications: the wire length has a different effect on the timing performance, the chip area is the major factor determining fabrication cost, and the number of inter-chip contacts affects the complexity of the 3-D fabrication process. In our implementation, we select λ and γ so that the first two terms can be roughly equal. Thus the floorplanner puts almost equal efforts to optimize wire length and chip area. The value of μ is chosen to reflect the relative "cost", e.g., footprint, of inter-chip contacts.

To consider the thermal effects, the cost function is extended by including another component, maximum on-chip temperature difference. The new cost function is as follows:

$$\begin{aligned} \text{Cost} = {} & \lambda \cdot \text{wire\_length} + \gamma \cdot \text{chip\_area} \\ & + \mu \cdot \text{num\_inter-chip-contacts} \\ & + \beta \cdot \max\{\text{on-chip-temp-diff}\} \end{aligned} \quad (5.13)$$

Since there are two device layers in the target implementation, we actually consider the sum of the maximum on-chip temperature difference of both layers. Clearly β determines how much effort the floorplanner will spend on optimizing

the temperature difference. In fact, the first three components in Equation (5.12) are correlated, i.e. optimizing one of them usually leads to better values for the remaining two. On the other hand, often a good temperature profile can only be achieved at the cost of a performance penalty, e.g., power-hungry modules along the critical path have to be distributed farther apart. The new cost component implies a design tradeoff between traditional design objectives and a temperature-wise objective.

During the simulated annealing process, neighboring solutions are explored by randomly perturbing the present solution. In our implementation, a random perturbation, or a random move, can be one of the three following ways: moving a block to another position, changing a block's aspect ratio, and swapping two blocks' positions. We allow inter-chip moves, which mean that a block can be removed from the BSG of one level and inserted into the BSG of another level. For the VLSI systems in Category 3, we allow another type of move, which is to assign a certain part of a module to another device layer. This is achieved by introducing a shadow block for each module. A module and its shadow block are assigned to a difference device layer in the 3-D implementation and the area assigned between them can be randomly changed.

During the floorplanning process, when a new solution is generated, we analyze the temperature distribution by considering both dynamic and leakage power consumptions. Since an on-line thermal analysis would be too expensive in terms of CPU time, we instead utilize pre-characterized thermal analysis data. Given an intermediate solution, i.e. the assignment of blocks in the 3-D layout as well as the aspect ratios of two chips, the thermal distribution results could be derived by interpolating pre-computed analysis data as described in the previous sections.

5.3.4 Experimental results

In our experiments, we use the largest three MCNC benchmark circuits, ami33, ami49 and playout introduced by Kozminski[14] to evaluate the results of thermal-driven floorplanning. We assume a Category 2 setup, although our methodology can be straightforwardly applied to Category 3 applications. The modules given in the original benchmark are only associated with a relative size, i.e. no physical unit is specified. Meanwhile, there's no dynamic power consumption information available. To enable the thermal analysis, the module sizes are scaled so that the total area is 2 cm^2. Our experiments are setup to examine the thermal impacts of floorplanning for 3-D/2.5-D ICs fabricated in deeply scaled technologies. We assume that 3-D/2.5-D chips are realized using high leakage devices in these advanced technologies. Additionally, since we are targeting at high power designs such as general purpose microprocessors, graphic processing units (GPU) and network processors, we set a relatively high dynamic power of 150 W. As such, on-chip temperature gradient tends to be large and hence is important to control properly. It is assumed that the dynamic power is uniformly assigned to each module, i.e. each module has a dynamic power value of P_{base} = 150 W/#modules. As a result, smaller modules will have higher power densities. We further assume the first 4 modules consuming 3, 2, 4, and 3 times of P_{base}. Our results show that, without temperature consideration, an area/wire length optimized solution tends to induce a very bad temperature profile. For instance, the 3-D floorplan of Benchmark Playout has a maximum temperature of 180.78℃ and a temperature gradient of 151.67℃.

As we mentioned earlier, the choice of β in the cost function actually suggests a continuously distributed tradeoff between two design objectives, wire length

and temperature gradient. Taking benchmark ami49 as an example, the tradeoff is illustrated in Table 5.3, where the first row shows the choice β values and the second row lists the initial percentage of the temperature cost in the total cost. Under the area/wire length optimization mode ($\beta = 0$), a solution with good wire length is established. However, the maximum on-chip temperature is 110.29℃, while the temperature gradient is 81.93℃. When the value of β increases, we observe smaller temperature gradient at the cost of longer wire length. The tendency of the tradeoff between wire length and temperature gradient is illustrated in Fig. 5.10, where the horizontal-axis is the total wire length and the vertical-axis is the on-chip temperature gradient. In fact, our optimization engine is quite effective to reduce on-chip temperature gradient: Even with a relatively small $\beta = 1$, the resultant temperature gradient is only 36.6℃. With a relatively large temperature gradient, the temperature gradient can be tuned to as small as 12.1℃.

Table 5.3 2.5-D thermal-driven floorplans with different weighting factors for thermal cost

Scaling factor	20	10	7.5	5	2.5	1	0
Temperature cost/ Total initial cost	0.8593	0.4297	0.3611	0.2736	0.1585	0.0701	0.0000
Total chip area	71600368	65216060	66855012	65234484	63471856	59724924	54925472
Worst case wire length	22340	22144	21094	18896	21388	19876	18728
Total wire length	2755434	2449634	2159834	2179952	1992604	1680000	1595962
Maximum temperature difference	12.06	16.49	17.59	27.82	29.66	36.6	81.93
Maximum temperature	66.31	66.83	66.69	71.85	75.01	76.31	110.29

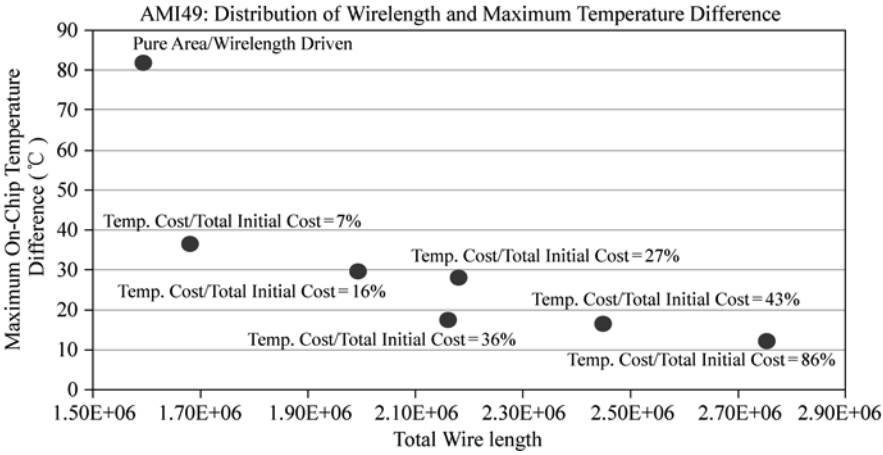


Figure 5.10 The distribution of wire length and temperature gradient

For a general thermal-driven 3-D floorplanning problem, we adopt the following rule to set the value for β. At the beginning stage, β is set to a value so that the cost corresponding to the maximum on-chip temperature difference is around 15% of total cost. Thus initially the optimization engine will spend major effort in reducing wire length. Meanwhile, only those moves with significant impacts on the temperature distribution would happen earlier. When solutions with relatively satisfying wire length and area have been found, the floorplanner will then spend most computational effort to reduce on-chip temperature difference.

Table 5.4 lists the experimental results of thermal-driven 3-D floorplanning. Compared with the area/wire length driven floorplanning results, the peak temperature reduces by 64.3℃, 52.3℃, and 112.5℃, respectively. The effectiveness of our thermal-driven floorplanner is shown in Fig. 5.11. The four drawings of temperature distributions are collected at four different stages in the optimization process. Obviously, at the beginning stage (Fig. 5.11(a)), the on-chip temperature is rather high and there exists a significant temperature gradient. Then both the

maximum on-chip temperature and temperature gradient goes down with time to a reasonable value at the final stage (Fig. 5.11(d)). Meanwhile, under most circumstances, the wire length penalty is fairly mild. Another key observation is that, the maximum on-chip temperature has also been effectively lowered, although it has not been explicitly considered in the cost function.

Table 5.4 3-D floorplans with and without thermal concern

Design	Measures of merit	Area & wire length driven	Area, wire length & thermal driven
AMI33	Total Area	1485386	2179324
	Worst Case Wirelength	5302	5358
	Total Wirelength	133994	171170
	Maximum On-Chip Temperture Difference (℃)	92.08	27.82
	Maximum On-Chip Temperture (℃)	122.92	73.55
AMI49	Total Area	54925472	63471856
	Worst Case Wirelength	18728	21388
	Total Wirelength	1595962	1992604
	Maximum On-Chip Temperture Difference (℃)	81.93	29.66
	Maximum On-Chip Temperture (℃)	110.29	75.01
Playout	Total Area	143097740	150188832
	Worst Case Wirelength	28562	30996
	Total Wirelength	9232726	10863240
	Maximum On-Chip Temperture Difference (℃)	151.67	39.14
	Maximum On-Chip Temperture (℃)	180.78	77.95

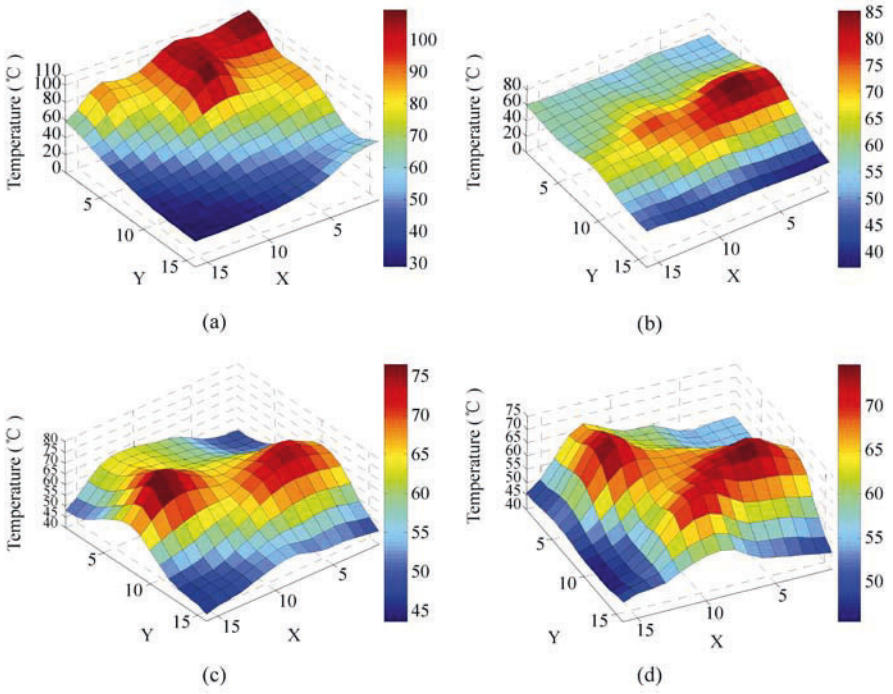


Figure 5.11 Temperature snapshots of the thermal driven floorplanning with Benchmark AMI49. Both the maximum temperature and the temperature gradient are reduced during the optimization (*see colour plate*)

5.4 Observations

In this chapter, we evaluate the feasibility of the 2.5-D integration on ASIC designs at the floorplan level. In the 2.5-D layout implantations, we observed sizable wire length reductions. Although the longest path may not be the one with the longest wire length, generally the wire length results, especially the top 1%－3% longest interconnects in large ASIC/SoC designs would have an important impact on the timing behavior[32]. Because of the unpredictability, long wires could

serious degrade the timing performance and/or increase design turnaround time. Accordingly, the experimental results showed that the 2.5-D implementations have a strong potential to achieve a better wire length distribution, which could be translated into improved performance and power consumption.

Our floorplanning tools could serve as the prototype for future 2.5-D system designs. A full-fledged 2.5-D floorplanning tool should be equipped with many new feature features described as follows.

Multiple-objective driven floorplan design Our current floorplan tools optimize a weighted sum of 4 components, total wire length, chip area, total number of inter-chip contacts, and maximal temperature difference on the chip surface. In the future, we might need to consider other factors, such as worst-case IR drop and data communication volume among blocks. For such a high-dimension solution space, an even distribution of optimization effort on multiple components might not lead to a good solution. An important extension is to develop a self-adapting optimization engine using techniques like particle swarm optimization[33].

Performance driven floorplanning for RF and analog circuits For RF and analog circuits, the system partition task is implicitly achieved within the floor-planning process. In other words, the floorplanner have to be able to allocate different circuit components to different layers to fully employ the advantages of 2.5-D integration. An important concern for this category of designs is the electromagnetic interference (EMI) noise among chips on different layers and in different technologies. Here the key requirement is to closely couple an electromagnetic analysis engine with the floorplanner. This way, at each stage of floorplan evaluation, important analog design metrics such as output Signal-Noise-Ratio (SNR)[34] can be assessed. The problem, again, is that a detailed electro-

magnetic analysis could be too expensive in terms of CPU time. The solution has to be developed by either developing fast evaluation techniques based on a coarser grain representation of the target design or interpolating pre-computed data. Besides EMI noises, the floorplanner has to also account for larger process variations and mismatches among different chips in a 2.5-D system.

References

[1] Magma Design Automation. The FixedTiming® methodology. [online]. Available: http://www.magma-da.com/c/@saRbjwTRmvDx6/Pages/fixedtiming.html.

[2] Cadence. Encounter digital IC design platform. [online]. Available: http://www.cadence.com/products/digital_ic/index.aspx.

[3] R. Otten. Efficient floorplan optimization. In: Proc. Int'l Conf. on Computer Design, 1983, pp. 499 – 502.

[4] D. F. Wong, C. L. Liu. Floorplan design of VLSI circuits. Algorithmica, 1989, pp. 263 – 291.

[5] H. Murata, K. Fujiyoushi, M. Kaneko. VLSI/PCB placement with obstacles based on sequence-pair. In: Proc. of Int'l Symposium on Physical Design, 1997, pp. 26 – 31.

[6] P.-N. Guo, C.-K Cheng, T. Yoshimura. An O-tree representation of non-slicing floorplan and its applications. In: Proc. Design Automation Conf., 1999.

[7] X. Hong, et al.. Corner block list: an effective and efficient topological representation of non-slicing floorplan. In: Proc. of Int'l Conf. Computer Aided Design, 2000, pp. 8 – 12.

[8] N. Sherwani. Algorithms for VLSI physical design automation. Kluwer Academic Publishers, 1999.

[9] M. Sarrafzadeh, C. K. Wong. An introduction to VLSI physical design. McGraw-Hill

Companies, 1996.

[10] NVidia Corporate, [online]. Available: http://www.nvidia.com/page/home.html.

[11] Y. Deng, P. Li. Temperature-Aware Floorplanning of 3-D ICs Considering Thermal Dependent Leakage Power. Journal of Low Power Electronics, Aug. 2006.

[12] S. Nakatake, et al.. Module placement on BSG-structure and IC layout applications. In: Proc. of Int'l Conf. Computer Aided Design, 1996, pp. 484 – 491.

[13] M. Huang, F. Romero, A. Sangiovanii-Vincentelli. An efficient general cooling schedule for simulated annealing. In: Proc. of Int'l Conf. Computer Aided Design, 1986, pp. 281 – 384.

[14] K. Kozminski. Benchmarks for layout synthesis. In: Proc. Design Automation Conf., 1991, pp. 265 – 270.

[15] F. Y. Young, Chris C. N. Chu, M. L. Ho. A unified method to handle all kinds of placement constraints in general non-slicing floorplan. In: Proc. IEEE Asia South Pacific Design Automation Conf., 2002, pp. 661 – 667.

[16] F. Y. Young, D. F. Wong. Slicing floorplans with range constraints. In: Proc. of Int'l Symposium on Physical Design, 1999, pp. 97 – 102.

[17] F. Y. Young, D. F. Wong. Slicing floorplans with pre-placed modules. In: Proc. of Int'l Conf. Computer Aided Design, 1998, pp. 252 – 258.

[18] P. Li, Y. Deng, L. Pileggi. Temperature-dependent optimization of cache leakage power dissipation. Proceedings of International Conference on Computer Design, 2005, pp.7 – 12.

[19] Y. Cheng, C. Tsai, C. Teng, S. Kang. Electrothermal analysis of VLSI systems. Kluwer Academic Publishers, 2000.

[20] T. Wang, C. Chen. 3-D thermal-ADI: a linear-time chip level transient thermal simulator. IEEE Trans. on Computer-Aided Des. Integrated Circuits System, 2002, Vol. 21, No. 12, pp. 1434 – 1445.

[21] T. Chiang, K. Banerjee, K. Saraswat. Compact modeling and spice-based simulation for electrothermal analysis of multilevel ULSI interconnects. Proceedings of International Conference on Computer-Aided Design, 2001, pp.165 – 172.

[22] H. Su, F. Liu, A. Devgan, E. Acar, S. Nassif. Full chip leakage estimation considering power supply and temperature variations. Proceedings of Intl. Symp. Lower Power Electronics and Design, 2003, pp. 78 – 83.

[23] P. Li, L. Pileggi, M. Asheghi, R. Chandra. Efficient full-chip thermal modeling and analysis. Proceedings of International Conference on Computer-Aided Design, 2004, pp. 319 – 326.

[24] C. Tsai, S. Kang. Cell-level placement for improving substrate thermal distribution. IEEE Trans. Computer Aided Design Integrated Circuits System, 2000, Vol. 19, No. 2. pp. 253 – 266.

[25] L. He, W. Liao, M. Stan. System level leakage reduction considering interdependence of temperature and leakage. Proceedings of Design Automation Conference, 2004, pp.12 – 17.

[26] K. Skadron, M. R Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, D. Tarjan. Temperature-aware microarchitecture. Proceedings of International Symposium on Computer Architecture, 2003, pp. 2 – 13.

[27] J. Cong, J. Wei, Y. Zhang. A thermal-driven floorplanning algorithm for 3D Ics. Proceedings of International Conference on Computer-Aided Design, 2004, pp.306 – 313.

[28] J. Cong, Y. Zhang. Thermal-driven multilevel routing for 3-D Ics. Proceedings of Asia South Pacific Design Automation Conference, 2005, pp.121 – 126.

[29] B. Goplen, S. S. Sapatnekar. Efficient thermal placement of standard cells in 3D ICs using a force directed approach. Proceedings of International Conference on Computer-Aided Design, 2003, pp.86 – 90.

[30] P. Li. Critical path analysis considering temperature, power supply variations and temperature induced leakage. Proceedings of 2006 IEEE Intl. Symp. Quality Electronic Design, 2006.

[31] Y. Saad. Iterative methods for sparse linear systems. SIAM, 2003.

[32] S. R. Mraz. Dissecting the PS3 'Cell' architecture. PS3Insider. [online]. Available: http://www.ps3insider.com/modules.php?name=Content&pa=showpage&pid=3.

[33] J. Kennedy, R. C. Eberhart. Swarm Intelligence. Morgan Kaufmann, 2001.

[34] J. Williams. The art and science of analog circuit design. Newnes, 1st Edition, 1998.

6 Placement for 2.5-D Integration

Yangdong (Steve) Deng

Institute of Microelectronics, Tsinghua University
Beijing 100084, P. R. China, dengyd@tsinghua.edu.cn

Wojciech P. Maly

Electrical and Computer Engineering Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891 USA, maly@ece.cmu.edu

Abstract This chapter covers the placement solutions for 2.5-D/3-D integrated circuits. Based on a partition technique, our placement techniques could handle VLSI circuits consisting of both standard cells and macros. The detailed result analysis justifies the potential of the 2.5-D integration approach to improve system performance and lower interconnect power consumption.

Keywords 2.5-D integration, placement, partition, mixed layout, standard cell, macro, wire length, inter-chip contact.

As we have proved the potential of the 2.5-D integration scheme at the floorplan level in the last chapter, it's appealing to investigate the feasibility at the

placement level, which implies a fine-grain partitioning of a VLSI system in a 2.5-D layout space. By comparing a design's monolithic and 2.5-D implementations at the placement level in terms of wire length distribution, in this chapter we assess the feasibility of the 2.5-D integration scheme at a finer abstraction level.

The goal of the monolithic cell placement problem is to embed a standard cell netlist on a two-dimensional plane to optimize certain design objectives. Before middle 1990s, the placement problem usually assumes a variable-die model, in which the number of routing tracks in a channel (i.e. the space between two neighboring cell row) and hence the total chip area could vary[1]. Thus, the optimization objective is to minimize both chip area and total wire length. Recently, the fix-die model has become dominant. Under such a context, the target layout area is fixed before placement and therefore the optimization objective is to minimize wire length. There is already a huge body of work focusing on the circuit placement problem. Interesting readers please refer to [2 – 9] for representative results.

The objective of the 2.5-D placement problem is to map a cell netlist (pure standard cell or mixed macro/standard cell) to unique positions in a layered space as illustrated in Fig. 6.1. The inter-chip contacts are assumed to be placed on top of the chip with no need to consume substrate area. We need to differentiate two scenarios: hierarchical and flattened design styles. In a hierarchical design set up, after the floorplanning step, cells in a block need to be placed. As mentioned in the last chapter, a random-logic based block could be split into two chips. The 2.5-D placement problem is to assign the cells within such a block to unique positions on two chips. On the other hand, in a flattened design style, the 2.5-D placement problem is to place both standard cell macros onto stacked chips.

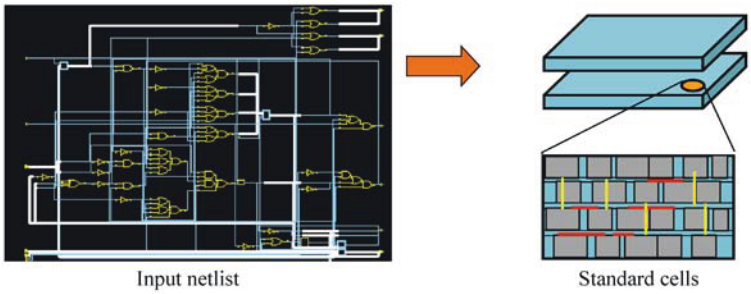


Figure 6.1 2.5-D placement problem (*see colour plate*)

In the following sections of this chapter, we studied the 2.5-D placement problem under the above mentioned three formulations: pure standard cell designs with inter-chip contacts consuming substrate area, pure standard cell designs with inter-chip contacts on top of die surface, and mixed standard cell and macro designs corresponding to a flattened design style.

6.1 Pure Standard Cell Designs

In this section, we consider the second scenario of 2.5-D placement, where a hierarchical design style is applied and the inter-chip contacts can be placed above the top-level metal layer.

Our 2.5-D placement tool is extended from Capo[1], which is a bi-partitioning based placement framework. The idea of bi-partitioning based placement is to recursively cut the input circuit into two parts so that the number of nets crossing partitioning boundary is minimized. With each cut, the layout region associated with the input circuit is divided into two smaller ones accordingly. The resultant two parts of the original circuit are then assigned into the two sub-regions. The process is repeated until every circuit only contains one cell, which means every

cell has been mapped to a unique position on the layout. The following list is the pseudo code of a typical partitioning based placement algorithm.

```
Bipartition Based Placement
   Input: netlist N
Output: cell location cellLoc[]
   Variables: Q = FIFO queue storing the subnetlists to be partitioned
   Subroutines:
   bipartition(Ni); //do FM partitioning on netlist Ni
   extractSubnetlist(Ni, N); //form subnetlist from local and global information
   Q.enqueue(Ni); //insert netlist Ni in the end of queue
   Q.dequeue(); //return the netlist at the head of queue
   Center(Ni); //return the center location of the layout area that contains
     subnetlist Ni
Begin
   NewN ← N;
   Q.enqueue(N);
   While Q is not empty Loop
                currN ← Q.dequeue();
                (N0, N1) ← bipartition(currN);
   For each i = 0 to 1 Loop
   extractSubnetlist(Ni, N);
   Forall cell ∈ Ni Loop
                                cellLoc[cell] ← center(Ni);
   If (number of cells in Ni > 1) Then
   Q.enqueue(Ni);
   Endif
   Endfor
           Endfor
   Endwhile
   return cellLoc[];
End
```

6.1.1 Placement Techniques

In the framework of partition based placement, at every stage of the monolithic

placement process, a circuit to be partitioned is associated with a planar region of the target layout. The circuit along with the layout region is designated as a block by Caldwell et al.[1]. In the 2.5-D layout, a circuit is associated with a bounded space consisting of two or more layers of layouts lying on different chips in the 2.5-D stack. We define such a space as a super-block. For the 2.5-D placement problem, a recursive partitioning procedure is carried out on the super-blocks. The process can be explained using the cube model illustrated in Fig. 6.2. For a 2.5-D system consisting of two levels of chips, cells should to be mapped to unique positions on either top or bottom surface of the cube after the placement process. Initially cells are assigned to the inner space of the cube model and not assigned to a specific surface. At a given stage, we bi-partition the netlist in a super-block and assign each partition to a different device level. We call this process as the vertical partitioning procedure. After this step, the placement process is continued just as in the 2-D placement problem.

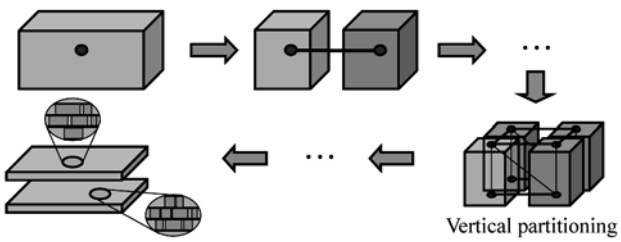


Figure 6.2 2.5-D placement process

During the vertical partitioning procedure, an essential issue is that the partition results should match the capacity of inter-chip communication resource. On one hand, the number of inter-chip contact on a given layout is determined by the interconnection technology. If the number of crossing-chip nets is beyond the

capacity of 2.5-D interconnects, some nets have to be routed in detour or can be even un-routable. On the other hand, if the number of crossing-level nets is too small, the potential of 2.5-D technology is not fully exploited.

In our current implementation, the vertical partitioning procedure is activated when the surface of a super-block can accommodate about $50-100$ inter-chip contacts because we found it's easier to match the demand and supply of inter-chip contact for the circuit at this scale. An effective means to control the number of inter-chip nets (i.e. nets with terminals in both levels of chips) during vertical partitioning is through balance tolerance, which is the maximum allowed cell area difference between the two partitions. Generally speaking, looser tolerance means less number of cuts since it allows more flexibility. During vertical partitioning, we initially set very tight tolerance, e.g., with 1%, because we want the two resultant partitions have roughly identical areas. Here the number of nets being cut is not a major concern as long as it's under the capacity of 2.5-D interconnects. After the first run, if the number of inter-chip nets is too large, we perform re-partition using a looser tolerance. This process is repeated until a satisfactory partition is derived. In our experiments, we always achieve convergence in less than three iterations.

An effective means to control the number of inter-chip nets (i.e. nets with terminals in both levels of chips) during vertical partitioning is through setting a proper balance tolerance, which is the maximum allowed difference between the total cell areas of two partitions. Generally speaking, looser tolerance leads to a smaller number of nets being cut due to the larger flexibility, while tighter tolerance suggests better control over the total cell areas in the resultant partitions. During vertical partitioning, we initially set very tight tolerance, e.g., within 1%, because we want to well manage the total cell areas of two resultant partitions. Here the

number of cut nets is not a major concern as long as it's under the capacity of 2.5-D interconnects. After the first run, if the number of inter-chip nets is too large, a re-partition is performed under a looser tolerance. This process is repeated until a satisfactory solution is achieved. In our experiments, we always achieve convergence in at most two iterations.

6.1.2 Benchmarks and Layout Model

Our standard cell benchmarks are from three sources: Sun Micro's processor benchmark suite[10], UCLA Dragon benchmark suite[11], and MCNC benchmarks[12]. These benchmarks have very diverse functionalities and complexities. Sun Micro benchmarks listed in Table 6.1.A are typical CPU circuits such as integer unit, float-point unit, memory management unit, and large register file. They are delivered

Table 6.1 Placement benchmarks
A. PicoJAVA and SPARC benchmark

Design	# Standard Cells	# Nets
icu	14222	9935
rcu	14393	10676
ex	21320	15594
fpu	24561	23347
iu	53854	42514
miu	28524	28498
mregfile	30114	29956
fpufpc	36922	36793

B. Dragon placement benchmark

Design	# Standard Cells	# Nets
IBM01	12282	13056
IBM02	19321	19291
IBM03	22207	26104
IBM04	26633	31328
IBM05	29347	29647
IBM06	32185	34935
IBM07	45135	46885
IBM08	50977	49228
IBM09	51746	59454
IBM10	67692	72760
IBM11	68525	78843
IBM12	69663	75157
IBM13	81508	97574
IBM14	146009	150262
IBM15	158244	183684
IBM16	182137	188324
IBM17	183102	186764
IBM18	210323	201560

C. MCNC placement benchmark

Design	# Standard Cells	# Nets
golem3	100312	144950

as Verilog HDL code and synthesized by us using Synopsys Design Compiler[13]. Circuits in the UCLA Dragon benchmark are adapted from different IC designs of IBM[14]. Characteristics of these circuits are shown in Table 6.1.B. We also used a MCNC benchmark circuit, golem3, because it has a relatively large complexity (Table 6.1.C).

In our placement experiments, we assume a fixed-die, over-the-cell routing model. Thus, the layout area of a design is the footprint of all cells plus 10% free space. The 2-D layout is mapped to a 2.5-D layout consisting of two stacking chips with equal area. All layouts have a square shape. Thus, the dimension of two chips in 2.5-D system is that of the corresponding 2-D layout scaled by 0.707. For every benchmark circuit, we generate both monolithic (2-D) placement and 2.5-D placement.

6.1.3 Evaluation of Vertical Partitioning Strategies

As mentioned before, we would perform the vertical partitioning procedure when the number of cuts matches the available capacity of vertical communication resource. In our experiments, we actually test the 2.5-D placement problem under two extreme cases: 1) inter-chip contacts have a very large pitch (e.g., similar to the pitch of flip-chip bumps) and the vertical partitioning has to be preformed at the first stage of the partition based placement process; 2) inter-chip contacts have a very small pitch and the vertical partitioning can be preformed at the finest level. Figure 6.3 compares the wire length reductions (compared with monolithic solutions) in the 2.5-D placements between these two cases using 5 randomly picked benchmark circuits.

In Fig. 6.3, “VP” is the short for vertical partitioning. The results demonstrate

that vertical, inter-chip interconnects do help reduce wire length. Actually, the smaller is the pitch, the larger is the number of inter-chip contacts allowed to be introduced, and thus the shorter wire length.

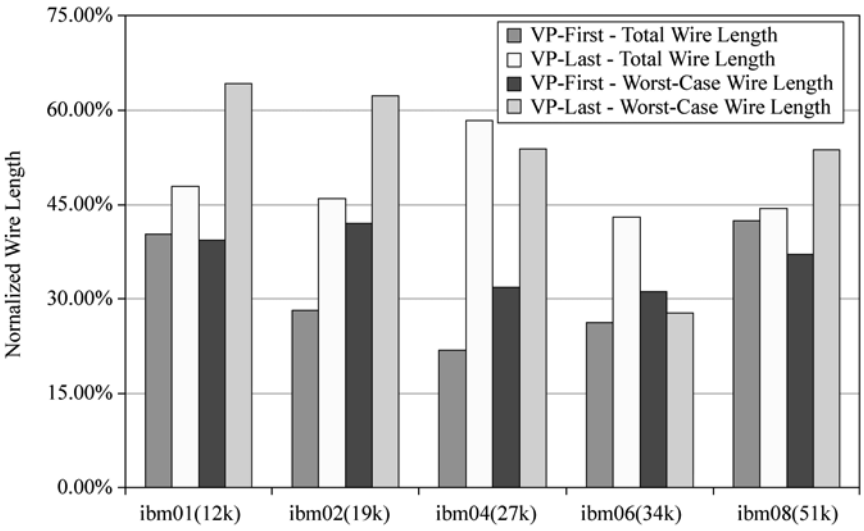


Figure 6.3 Wire length reductions vs. vertical partitioning

6.1.4 Wire length scaling

In the partitioning based placement process, nets in a netlist may be cut (by "cut" we mean a net has cells in different partitions) at different stages. Generally, nets containing more loosely connected cells (which have fewer interconnections among them) are cut earlier and they tend to have longer wire length. In other words, nets cut at different stages will demonstrate different patterns in the reduction of wire length. Figure 6.4 is an illustration of this effect. In this drawing, suppose we have both 2-D and 2.5-D layouts for the same design. If the layout dimension of

monolithic layout is a, then the layout dimension of 2.5-D placement is 0.707a. Suppose block A in the monolithic layout is mapped to super block A' in the 2.5-D layouts and the vertical partitioning is performed on A'. It bears mentioning that the dimension of A' is correspondingly scaled by 0.707. Let's consider Net 1, which is already cut when block A is going to be partitioned. Since the size of all blocks crossed by net 1 is scaled down by 0.707, the wire length of Net 1 in the 2.5-D layout will be shrunk by 0.707 accordingly. On the other side, the wire length of nets inside A like Net 2, will keep unchanged. Of course, partition results will not be exactly identical in monolithic and 2.5-D placements, but the above analysis holds true statistically.

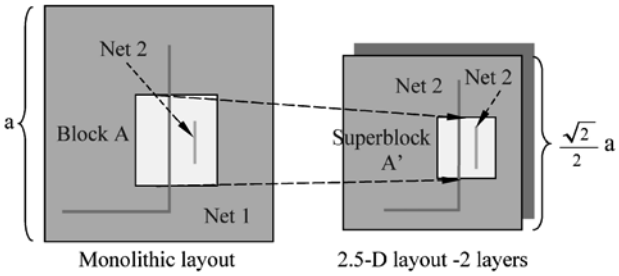


Figure 6.4 Monolithic and 2.5-D placements for the same design

In Fig. 6.5, we pick up the five biggest benchmark circuits, which have the same number of partition levels. We group the nets in each design by the partition level at which they are first cut during the placement process. The total wire length reductions of each group in the 2.5-D placements compared with the 2-D equivalents are shown in Fig. 6.5. For these circuits, the 2.5-D partition occurs at cut level 11 – 13. At the final level (level 19), the placer calls a branch-and-bound end-case partitioning procedure to complete the detailed placement. Observing

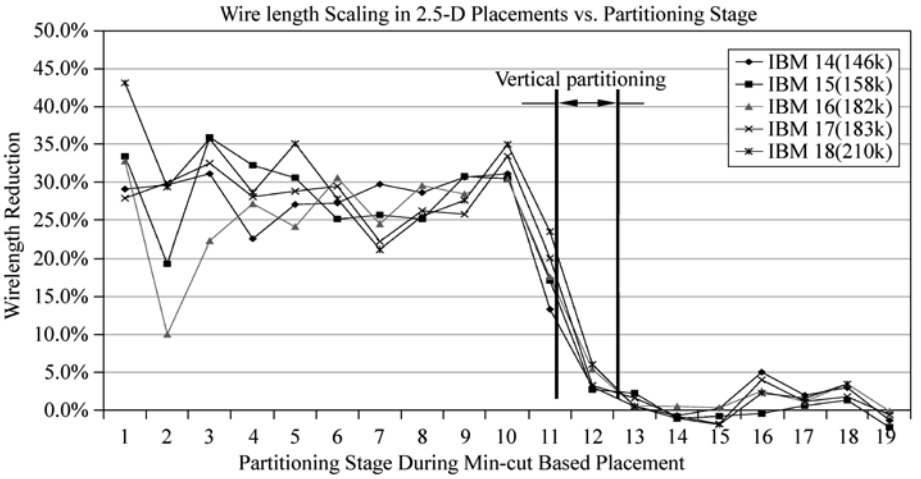


Figure 6.5 A profile of wire length reduction

Fig. 6.5, we can find that the wire lengths of nets that are already cut before 2.5-D partitioning are reduced by roughly 30%. Meanwhile, the nets cut after the vertical partition procedure usually have their length unchanged. The results reveal the potential of the 2.5-D layout to reduce wire length. Besides, the above analysis and results also imply a way to predict the wire length reduction using monolithic placement data.

In a typical netlist, there exist some nets with large number of fan-outs (e.g., >15, up to several hundreds), which are usually clock and reset signals. In our experiments on PicoJAVA and SPARC benchmark circuits[10], these nets have the longest wire length but do not appear in the critical path of a design. Hence, reducing wire length of these nets will not directly lessen system delay but will improve other design metrics such as clock skew, power consumption and routability. According to previous analysis, it can be predicted that theoretically the wire length values of these nets will be shrunk by 0.707 in the 2.5-D placements. In Table 6.2 we compare the wire length results of the large fanout nets measured

from 2-D and 2.5-D placements. The worst-case wire lengths of these large-fanout nets are all reduced by around 29.3%. In fact, the clock and reset pins will be placed almost everywhere on the chip, so the wire length of these nets are proportional to chip dimension. As we mentioned earlier, the dimension of 2.5-D layout is 0.707 of the dimension of 2-D layout. Consequently, we can conclude that the wire length reduction of high-fanout nets is contributed by chip dimension shrinking.

Table 6.2 Worst-case wire length reduction for nets with large fan-out

Design	Longest wirelength (2-D)	Longest wirelength (2.5-D)	Reduction
icu	308932	220342	28.7%
rcu	299670	211945	29.3%
ex	385973	272694	29.3%
fpu	467628	329740	29.5%
iu	648055	458127	29.3%
miu	531126	373847	29.6%
mregfile	532799	377235	29.2%
fpufpc	566166	399219	29.5%

6.1.5 Wire length reduction

Since the nets with large fan-out number will shrink by a constant factor and do not affect system performance directly, we only consider nets with less than 15 pins in the following discussion. In the results reported in the rest of this paper,

we will not take into account the wire length contributed by high fan-out nets in our study on wire length distribution.

For all benchmark circuits listed in Table 6.1, the 2.5-D solution always has shorter wire length than the monolithic solution. Thus in Fig. 6.6, we list the potential reductions for both total wire length and worst-case wire length in the 2.5-D placement corresponding to data listed in Table 6.3. On average we can achieve significant reductions of 24.8% and 27.65% in total wire length and worst-case wire length, respectively.

We can further compare 2-D and 2.5-D schemes in terms of wire length distribution. Figure 6.7 shows wire length distributions in both 2-D and 2.5-D placements of a well known MCNC benchmark circuit with 100k gates. Compared with its 2-D counterpart, 2.5-D wire length distribution is "compressed" into the left, which means there are substantially fewer global and semi-global wires. Since global wires usually determine the total signal delay, smaller number of them and shorter wire length imply 2.5-D system's considerable potential for

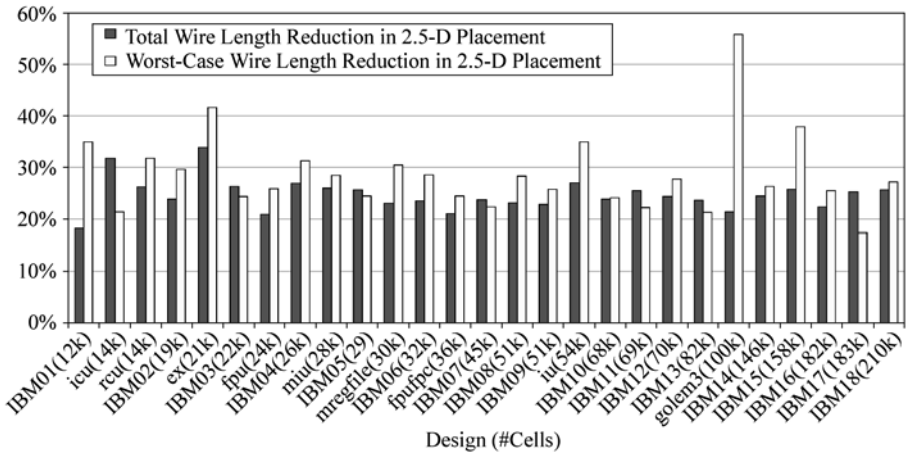


Figure 6.6 Wire length reductions of standard cell placement

Table 6.3 Wire length comparison of standard cell placements

Design	Total wire length (2-D)	Total wire length (2.5-D)	Reduction	Longest wire length (2-D)	Longest wire length (2.5-D)	Reduction
icu	4.06759E+07	2.77280E+07	31.8%	95910.7	75360.1	21.4%
rcu	3.93696E+07	2.90408E+07	26.2%	108478	73949.9	31.8%
ex	7.96698E+07	5.26400E+07	33.9%	137573	80247.8	41.7%
fpu	6.82364E+07	5.39649E+07	20.9%	150772	111659	25.9%
iu	1.87736E+08	1.36806E+08	27.1%	236734	153989	35.0%
Miu	1.54141E+08	1.14049E+08	26.0%	237650	169804	28.5%
Mregfile	1.09394E+08	8.41744E+07	23.1%	227715	158299	30.5%
fpufpc	1.32108E+08	1.04193E+08	21.1%	148956	112457	24.5%
IBM01	3.03150E+07	2.47958E+07	18.2%	67713.4	43981.9	35.0%
IBM02	9.25819E+07	7.04567E+07	23.9%	105188	73936.9	29.7%
IBM03	8.13335E+07	5.98733E+07	26.4%	100067	75768.8	24.3%
IBM04	1.32613E+08	9.69875E+07	26.9%	109927	75555.7	31.3%
IBM05	1.26721E+08	9.43262E+07	25.6%	134824	101737	24.5%
IBM06	9.71956E+07	7.43888E+07	23.5%	135304	96570	28.6%
IBM07	2.04216E+08	1.55540E+08	23.8%	132125	102555	22.4%
IBM08	1.64719E+08	1.26468E+08	23.2%	121286	86920.9	28.3%
IBM09	1.81519E+08	1.40153E+08	22.8%	134641	99949.9	25.8%
IBM10	3.47150E+08	2.64059E+08	23.9%	166658	126446	24.1%
IBM11	3.06845E+08	2.28542E+08	25.5%	144663	112518	22.2%
IBM12	4.36614E+08	3.30509E+08	24.3%	228494	164969	27.8%
IBM13	3.30063E+08	2.52049E+08	23.6%	173786	136702	21.3%

(Continued)

Design	Total wire length (2-D)	Total wire length (2.5-D)	Reduction	Longest wire length (2-D)	Longest wire length (2.5-D)	Reduction
IBM14	7.68137E+08	5.79874E+08	24.5%	285193	209904	26.4%
IBM15	8.37679E+08	6.21418E+08	25.8%	303150	188243	37.9%
IBM16	1.02283E+09	7.95108E+08	22.3%	329106	245067	25.5%
IBM17	1.43706E+09	1.07380E+09	25.3%	319515	264052	17.4%
IBM18	8.36957E+08	6.22105E+08	25.7%	318514	231857	27.2%
Average			24.8%			27.65%

higher performance. Meanwhile, semi-global wires contribute 60%－70% of total wire length according to our experiments. Thus semi-global wires tend to determine system power consumption, which is another major concern for today's VLSI system, especially for mobile devices. Consequently, the significantly reduced number of semi-global wires in the 2.5-D solution suggests potential of substantial power saving.

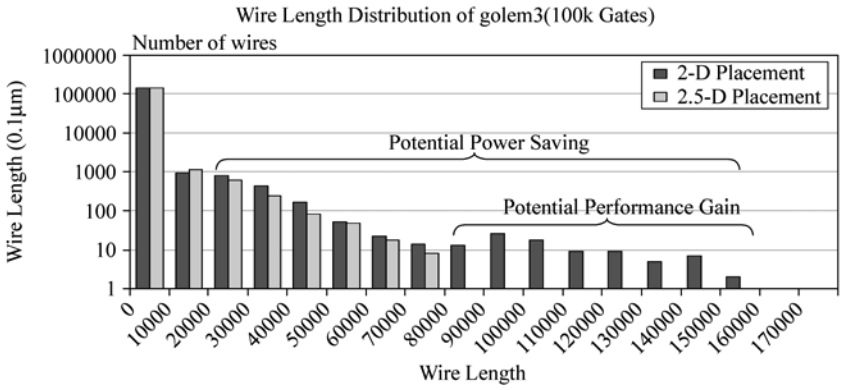


Figure 6.7 Wire length distribution of one design

We actually calculate the interconnection dynamic power consumption using the estimated wire length results for the IBM benchmark circuits. The results are illustrated in Fig. 6.8. On average, the 2.5-D solutions consume 26.5% less power on the wires.

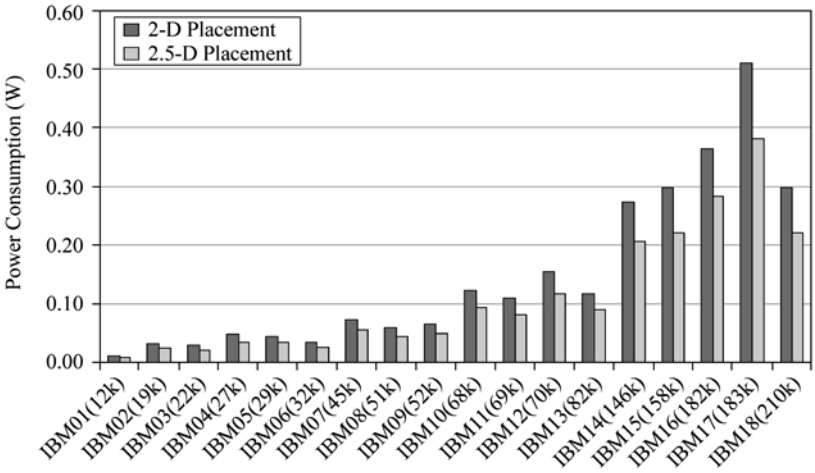


Figure 6.8 Interconnect power comparison—2-D and 2.5-D solutions

6.1.6 Wire Length vs. Inter-Chip Contact Pitch

Here we'll give an example of analyzing the tradeoff between the pitch of inter-chip contacts and the total wire length (which can be a measure of performance). The four benchmark circuits in Fig. 6.9 are subsystems (e.g., integer unit, float point unit and its controller, etc.) of Sun Micro's PicoJAVA and SPARC benchmark suites[10]. The benchmarks are distributed as RTL Verilog code. They are synthesized in a 0.18 mm CMOS library by Synopsys Design Compiler[13] and then placed by our 2.5-D placer into two stacked chips. Figure 6.9 shows the normalized wire

length of the benchmark circuits in their 2.5-D implementations (For each circuit, assuming its wire length is 1 when the pitch of inter-chip contact is 20 μm×20 μm). A general trend in Fig. 6.9 is that the wire length becomes shorter when the pitch of inter-chip contacts is smaller. On the other hand, different designs have varied sensitivity on the pitch value. For benchmark fpufpc, when pitch size is reduced from 8 μm×8 μm to 5 μm× 5 μm, total wire-length can be reduced by ~20%. On the other hand, for the remaining three benchmark circuits, it does not gain much to make the pitch smaller than 10 μm× 10 μm.

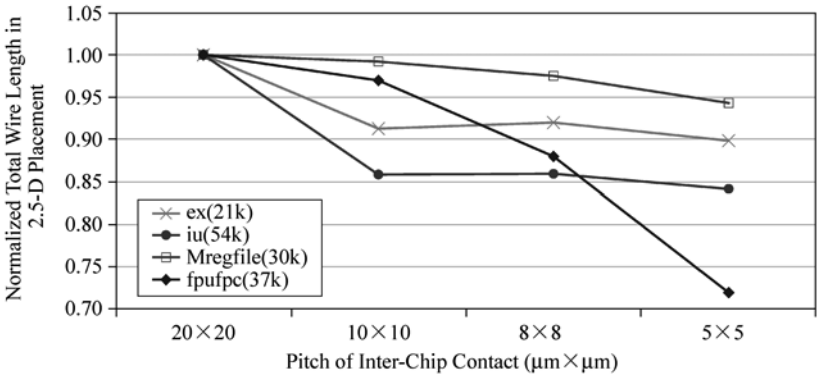


Figure 6.9 Wire length vs. pitch of inter-chip contact pitch

6.2 Mixed Macro and Standard Cell Designs

The original placement framework of Capo only supports pure standard cell layout. However, today typical VLSI designs contain certain number of macros, such as embedded memory and IP blocks. In a flattened design flow, placement engine has to be able to place macros and standard cells simultaneously. Accordingly, we enhance Capo with the capability to handle mixed macro/standard cell layout.

We then extend the placement framework into the 2.5-D scenario.

The layout style using mixed macro/standard cells is proper for VLSI design containing IP blocks and embedded memories to be aggressively optimized in a flattened manner. In this layout style, a small number of large macros are scattered in a "sea" of standard cells. For this layout style, a natural question is whether the cell area variability will impair the efficiency of 2.5-D integration. Therefore, we enhance our placement framework with capabilities of handling macros.

Table 6.4 lists the mixed benchmarks we used. These circuits are adapted from ISPD98[14] partition suite. The circuits are originally named from IBM01 to IBM18. Except circuit IBM05, 17 out of these 18 circuits consist of large number of cells with small weight (smaller than 10000) and a few cells with especially large weight (far larger than 10000). Consequently, we treat those large cells as macros and assign them with randomly chosen aspect ratios. For small cells we set the cell width and height consistent with ST Microelectronics' 0.18 mm library[15].

Table 6.4 Mixed Layout Benchmarks

Design	# Standard Cells	# Macros	# Nets	Area% of All Macros	Area% by The Largest Macro
MIX01	12503	3	14111	7.1	6.37
MIX02	19330	12	19584	45.6	11.36
MIX03	22843	10	27401	40.6	10.76
MIX04	27212	8	31970	28.8	9.16
MIX06	32320	12	34826	43.6	13.56
MIX07	45628	11	48117	26.7	4.76
MIX08	51008	15	50513	35.2	12.10

(Continued)

Design	# Standard Cells	# Macros	# Nets	Area% of All Macros	Area% by The Largest Macro
MIX09	53055	55	60902	41.1	5.42
MIX10	68631	54	75196	62.2	4.80
MIX11	70097	55	81454	35.5	4.48
MIX12	70235	204	77240	54.8	6.43
MIX13	83610	99	99666	35.3	4.22
MIX14	147024	64	152772	9.9	1.99
MIX15	161166	21	186608	25.6	11.00
MIX16	182620	360	190048	45.9	1.89
MIX17	184646	106	189581	9.0	0.94
MIX18	210336	5	201920	4.2	0.96

6.2.1 Placement Techniques

Compared with pure standard cell placement problem, the difficulty of handling mixed macro/standard cell is how to efficiently remove overlap among cells and macros. Macros are large physical objects with arbitrary shape, and one macro usually interconnects with many small cells. As a result, moving one macro may change the solution structure of many small cells. On the other hand, small standard cells are more flexible and can accommodate the arbitrarily shaped space unoccupied by macros. Consequently, macros and standard cells should be placed concurrently in an interleaved manner.

In the benchmarks listed in Table 6.4, the largest area percentage of total chip area occupied by a single macro is below 14%, which is relatively small. Therefore, we believe we can still use the top-down partitioning technique in the placement if we carefully control the cut line during partitioning. Unlike the case of standard cell placement, where we can use the available tool for the monolithic problem, now we have to build both monolithic and 2.5-D placement tools. Again we developed this tool on the basis of UCLA's Capo placer.

For the intra-level placement problem, we start the placement process just as in the pure standard cell problem. When a block contains a macro consuming a certain percentage (now it is set as 1/3) of the total block area, we split the block to place the macro. A macro-oriented splitting will form three sub-blocks, a sub-block with a single macro and two blocks with multiple cells. The splitting direction (vertical or horizontal) is selected to make the resultant block with aspect ratio closer to 1. We check the average x and y coordinates of the macro's neighbors to determine to which sub-block the macro is assigned. The sub-block containing the macro has width (vertical splitting) or height (horizontal splitting) set exactly equal to the width or height of the macro. The position of the macro is then fixed. We try all four possible orientations for a macro and choose the one with minimum total wire length. Next the netlist in the block is partitioned assuming the macro is fixed in one partition. The resultant partition without macro is converted into a sub-block. The other resultant partition containing the macro is split again. Smaller cells are associated with the half layout not occupied by the macro to form a new sub-block. The two sub-blocks with multiple smaller cells will be further partitioned and placed.

The splitting process is illustrated in Fig. 6.10. In this drawing, suppose we are going to split a block, which contains a macro. Because neighbors of this macro

are all located on the right and bottom side, we assign the macro to the right bottom corner of the current block. Then we partition the netlist assuming the macro is fixed in the lower half. After partitioning, we will have a macro sub-block and other two sub-blocks, one located in the upper half and the other in the left bottom corner. To make the 2.5-D extension, we start the vertical partition when a block contains 500 standard cells or a macro consumes more than 1/6 of total block area. During the vertical partition, we set very tight balance tolerance so that we can accurately control the block size for the future placement process.

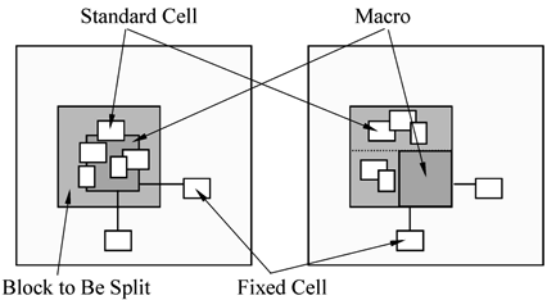


Figure 6.10 Block splitting during mixed placement

6.2.2 Results and Analysis

In Table 6.5 and Fig. 6.11, we list the wire length comparison between 2-D and 2.5-D mixed placements. We observe 21.2% and 28.9% reductions in total wire length and worst-case length, respectively.

The above results suggest that the 2.5-D integration scheme again leads to significant saving when applied to mixed standard cell/macro layout style. In other words, the cell area variance will not affect the feasibility of 2.5-D integration.

Table 6.5 Wire length characteristics of mixed placement

Design	Total wire length (2-D)	Total wire length (2.5-D)	Reduction	Longest wire length (2-D)	Longest wire length (2.5-D)	Reduction
IBM01	7.92066e+07	5.73135e+07	27.6%	140962	101604	27.9%
IBM02	1.54282e+08	1.36545e+08	11.5%	206103	132630	35.6%
IBM03	2.64642e+08	1.93691e+08	26.8%	221095	179942	18.6%
IBM04	2.67487e+08	2.27015e+08	15.1%	174867	117187	33.0%
IBM06	1.47301e+08	1.13644e+08	22.8%	168769	112290	33.5%
IBM07	3.42466e+08	2.60920e+08	23.8%	215115	141372	34.3%
IBM08	3.28108e+08	2.69640e+08	17.8%	224536	139083	38.1%
IBM09	4.40590e+08	3.56746e+08	19.3%	245552	186762	23.9%
IBM10	9.88970e+08	7.68081e+08	22.3%	495177	403773	18.5%
IBM11	6.29094e+08	5.46109e+08	13.2%	335956	187031	44.3%
IBM12	1.16658e+09	8.38296e+08	28.1%	354318	287869	18.8%
IBM13	6.99013e+08	5.98002e+08	14.5%	319335	232427	27.2%
IBM14	1.08321e+09	8.46040e+08	21.9%	318874	253982	20.4%
IBM15	1.49327e+09	1.16348e+09	22.1%	493873	282998	42.7%
IBM16	1.72296e+09	1.37654e+09	21.7%	461148	342712	25.7%
IBM17	1.90023e+09	1.34852e+09	29.0%	416505	282055	17.7%
IBM18	9.24663e+08	7.08821e+08	23.3%	323425	221093	31.6%
Ave.			21.22%			28.93%

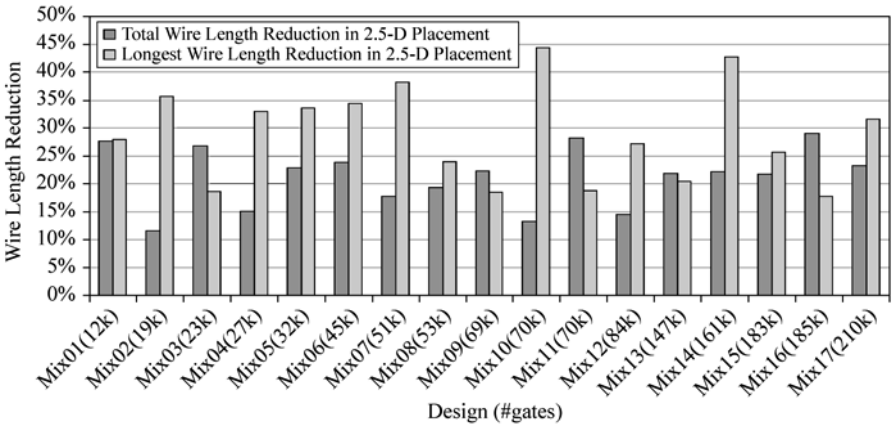


Figure 6.11 Wire length reductions of mixed placement

6.3 Observations

In this chapter, we evaluated the feasibility of 2.5-D integration at placement level design representations. Revisiting the classical placement problem under the 2.5-D integration paradigm, we developed efficient 2.5-D placement techniques by extending a partition-based placement framework. Our placement tools could handle both hierarchical and flattened design styles. A key feature is that our tools could optimize wire length under the constraint of inter-chip interconnection resource. A large number of ASIC applications in different design styles were excised with our tools. The experimental results proved that the 2.5-D implementations would promise improved wire length distribution and thus very likely performance improvement over the monolithic implementations. Combined with our floorplanning and placement results, we observed that 2.5-D integration allows a better wire length distribution in all three major layout design styles: block based design, hierarchical design, and flattened design.

Our 2.5-D placement tools could serve as the starting point to develop a full-fledged placement tools set for future 2.5-D/3-D ICs. Specifically, future 2.5-D placement tools have to be able to optimize multiple cost objectives in addition to the traditional wire length and critical path delay.

Inter-chip routability Besides the conventional routability issues, the inter-die wiring congestion can be caused by the mismatch between the routing demand for inter-chips contacts and the available routing resource. Such congested regions would lead to detoured nets. To solve the problem, the inter-chip routing demand and supply have to carefully modeled and updated during the placement process. If it's too difficult to maintain such information during the placement process, the 2.5-D placement can be followed by a routability driven migration process so that the demand can be lowered by moving the least number of place-able objects.

Hot-spot avoidance To guarantee the correct functionality of an IC, the heat generated by devices and wires must be effectively and efficiently removed. The 2.5-D placement tool has to generate solutions with a relatively even thermal map and low peak temperature. There has been a large body of work (e.g., [16,17]) proposed to solve the problem. For analytical based placers, the requirement for an even thermal can be formulated as a set of equations as constraints. On the other hand, under the context of partitioned based placement, a high power cell can be bloated to a certain extent so that it can be allocated with a larger white space and thus lower power density.

Timing optimization Generally we can expect better timing in a 2.5-D implementation because of the shorter wire length. In addition, the 2.5-D integration offers new opportunities to optimize timing performance during the placement stage. For instance, it would be very useful if we can migrate an existing

placement solution for better timing by folding a long timing path. Alternatively, for a long wire in a given chip, we can drop one or more buffers on the other chip to reduce the wire delay. The ability of buffer insertion "in the 3rd dimension" is important because many times a good buffer location could already be occupied when only one device layer is available.

Multiple user constraints It is very likely that a future 2.5-D placement tool has to handle varying user constraints. These constraints can be given in the form of clustering requirement. For instance, standard cells in a datapath element will have to be placed in a regular manner (either in a 2-D or 2.5-D organization). Another form of constraint is the bounding-box constraints. One example is that the power and current densities within a given region must honor a given specification to avoid reliability issues. In addition, when a process variation map is available, the cells on a critical path might need to be placed within a certain region.

References

[1] A. E. Caldwell, A. B. Kahng, I. L. Markov. Can recursive bisection alone produce routable placements? In: Proc. Design Automation Conf., 2000, pp. 477 – 482.

[2] C. Sechen, A. Sangiovanni-Vincentelli. The Timberwolf placement and routing package. IEEE J. Solid-State Circuits, Vol. SC-20, Apr. 1983, pp. 510 – 522.

[3] J. M. Kleinhans, G. Sigl, F. M. Johannes, K. J. Antreich. GORDIAN: VLSI placement by quadratic programming and slicing optimization. IEEE Trans. on Computer Aided Design, Vol. 10, No. 3, 1991, pp. 365 – 365.

[4] G. Sigl, K. Doll, F. M. Johannes. Analytical placement: a linear or a quadratic objective function? In: Proc. Design Automation Conf., 1991, pp. 427 – 432.

[5] H. Chen, C.-K. Cheng, N. -C. Chou, A. B. Kahng. An algebraic multigrid solver for analytical placement with layout based clustering. In: Proc. Design Automation Conf., 2003, pp. 794 – 799.

[6] A. E. Dunlop, B. W. Kernighan. A procedure for placement of standard-cell VLSI circuits. IEEE Trans. on Computer-Aided Design, Vol. CAD-4, No. 1, Jan. 1985, pp. 92 – 98.

[7] F. M. Johannes, H. Eisenmann. Generic global placement and floorplanning. In: Proc. Design Automation Conf., 1998, pp. 269 – 274.

[8] M. Wang, X. Yang, Majid Sarrafzadeh. Dragon2000: standard-cell placement tool for large industry circuits. In: Proc. of Int'l Conf. Computer Aided Design, 2000.

[9] M. R. Hartoog. Analysis of placement procedures for VLSI standard cell layout. In: Proc. Design Automation Conf., 1986, pp. 314 – 319.

[10] Sun Micro PicoJava Benchmark. [online]. Available: http://www.sun.com/processors/communitysource.

[11] Dragon Placement benchmark. [online]. Available: http://www.cs.ucla.edu/~xjyang/Dragon/.

[12] K. Kozminski. Benchmarks for layout synthesis. In: Proc. Design Automation Conf., 1991, pp. 265 – 270.

[13] Synopsys Inc.. Design compiler user guide. 1999.10.

[14] C. Alpert. The ISPD circuit benchmark suite. In: Proc. Int'l Symposium on Physical Design, 1998, pp. 80 – 85.

[15] STMicroelectronis. [online] Available: http://www.st.com.

[16] G. Chen, S. Sapatnekar. Partition driven standard cell thermal placement. Proc. of Int'l Conf. on Physical Design, 2003, pp. 75 – 80.

[17] B. Obermeier, F. M. Johannes. Temperature-aware global placement. Proc. of Asia South Pacific Design Automation, 2004, pp. 143 – 148.

7 A Road map of 2.5-D Integration

Yangdong (Steve) Deng

Institute of Microelectronics, Tsinghua University
Beijing 100084, P. R. China, dengyd@tsinghua.edu.cn

Wojciech P. Maly

Electrical and Computer Engineering Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891 USA, maly@ece.cmu.edu

Abstract In this chapter we propose a roadmap for the development of 2.5-D integrated VLSI systems. Beginning with relatively simple stacked memory chips, we envision that the 2.5-D paradigm will be gradually adopted by CPUs, graphic processors, mixed-signal systems, and extremely high-performance applications. The 2.5-D technology would unleash the power to build VLSI applications that are hard to imagine today.

Keywords 2.5-D Integration, stacked memory, CPU, graphics processing unit (GPU), DRAM, mixed-signal, image sensor, radar-in-cube.

In the previous chapters, the feasibility of 2.5-D integration scheme has been justified through extensive design case studies. With the maturation of fabrication,

testing, and design technologies, we believe 2.5-D ICs will open new paths to build future VLSI applications. It can be expected that new VLSI applications would be enabled by the 2.5-D paradigm. Moreover, existing applications would be ported to 2.5-D implementations for improved performance and/or reduced cost. As illustrated in Fig. 7.1, in this chapter we propose a roadmap with estimated time lines for the adoption of 2.5-D paradigm by major VLSI applications. In the remaining of this chapter, we will also discuss key design issues and efficient system architectures for these applications so that the full potential of 2.5-D integration can be realized.

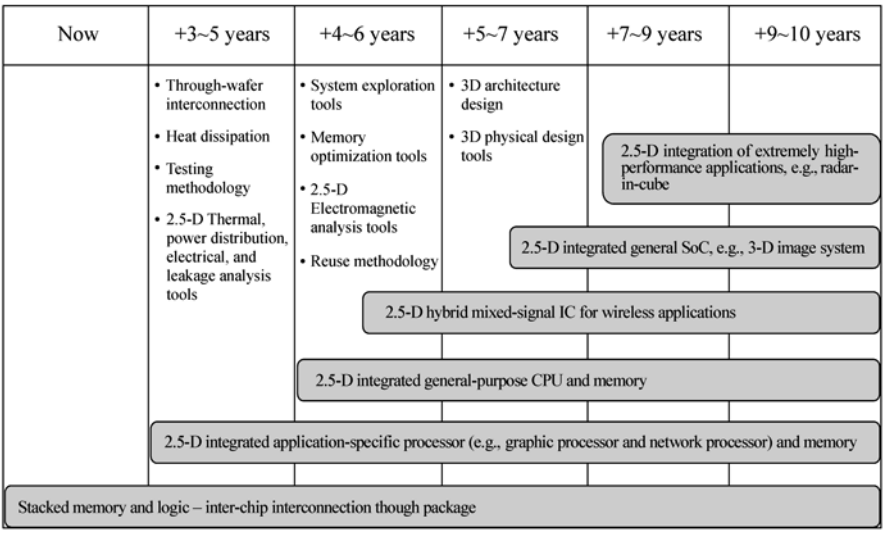


Figure 7.1 Road map for the development of 2.5-D ICs

7.1 Stacked Memory

The stacked memory, or so-called stacked chip scale package (SCSP) memory, which has been in the market for a couple of years[1–4], could be considered as

the 0th generation of 2.5-D integrated systems. In fact, there is no doubt that the stacked memory is an ideal solution to address two seemingly contradicting requirements for consumer electronic devices.

Diminishing Weight and Size Targeting a highly competitive market, today's consumer electronic devices have to satisfy stringent weight and size requirements. A typical cellular phone has to weigh less than 60 g and occupies no more than 50 cm^2 a PCB area (Just imagine such a phone would had a weight of 800 g and a PCB area of 150 cm^2 in 1986)[5].

Larger Memory Capacity The memory capacity in typical consumer devices has always been increasing. For instance, to support multi-mode (e.g., AMPS+ TDMA), multi-band (e.g., 900 and 1800 MHz) communication and PDA functionalities, a 3G wireless handset would need 128 Mb flash memory and 128 Mb DRAM[6]. Figure 7.2 shows the trend of Flash memory capacity in cellular phones. A salient trend is that the memory demand has outpaced the memory capacity that can be provided by pure scaling[7].

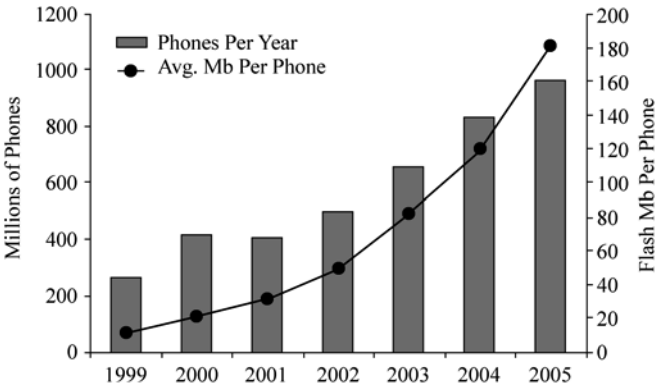


Figure 7.2 Flash memory capacity in cellular phones (adapted from[5])

7.2 DRAM Integration for Bandwidth-Demanding Applications

The performance gap between logic devices and memory has long been a bottleneck for modern networking and multimedia applications. It can be foreseen that, however, future broadband application would pose even more challenging demand for memory bandwidth. In the remaining of this sub-section, we will first review the constantly increasing demand for memory bandwidth and then discuss how to address the problem under the 2.5-D integration paradigm.

Here we choose NVidia's graphic processor units (GPU) to illustrate the trend in memory bandwidth demand. Figure 7.3 shows the peak memory bandwidth of each generation of GPUs released by NVidia's. Within each generation of its GPU portfolio, actually NVidia's would provide multiple derivative chips including both full-fledged versions and simplified versions targeting different markets segments. To make a consistent comparison, in Fig. 7.3 we only consider the full-featured, "flagship" products ased in the high-end applications. The data for this drawing were compiled from NVidia's website[8] and two other data-analysis websites[9,10], providing detailed technical reviews for video cards.

It can be observed from Fig. 7.3 that the requirement for peak memory bandwidth for NVidia GPUs has increased from 0.53 GB/s to 35.2 GB/s, or a factor of 66.7, in a period of 10 years beginning from 1994 (NVidia introduced its first GPU, NV1, in 1994). This momentum has to be continued with the adoption of advanced graphic features such as the 128-bit floating-point color depth and DVD quality real-time computer games.

Although it already demonstrates many key advantages, the SCSP memory is of course not a fully 2.5-D integrated system yet. Most of all, the inter-chip

communication has to be routed through I/O pads located on the chip periphery (as shown in Fig. 7.4). In other words, the stacking styled assembling is only meant to reduce weight and size, but not to improve system performance. Beginning from the next section, we will discuss more aggressive adoption of the 2.5-D scheme.

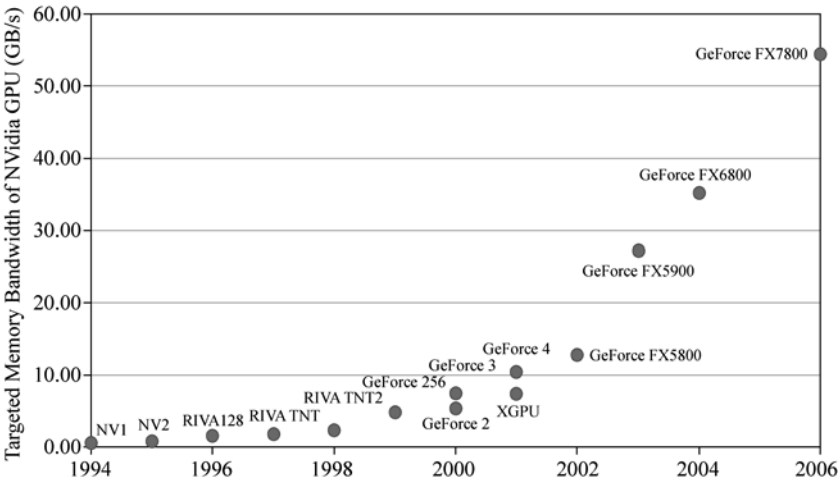


Figure 7.3 Peak memory bandwidths of major NVidia GPUs

(a) The Concept of Stacked Chip Scale Package

(b) Photo of Flash chips in Stacked Chip Scale Package

Figure 7.4 Intel's wire-bonded stacked Chip Scale Packaged flash memory (courtesy of Intel Corporation)[4]

Another key observation is that the bandwidth demand has already outpaced the bandwidth improvement through process scaling and DRAM technology innovations. This effect is illustrated in Fig. 7.5 in which both the available memory clock frequency (the area curve in the darker color) and the peak memory data rate (the bars in the lighter color) are normalized to their 1994 values (66 MHz and 0.53 GB/s, respectively). Apparently, within the same period the memory clock has only improved by a factor of 16.7. To compensate for the insufficient memory clock frequency, the remaining improvement factor of 4 is realized by increasing the width of memory bus from 64 bits and 256 bits. However, the problem is that, a larger number of off-chip I/O pins will be required, which will pose significant challenges for the package design. A common practice to overcome the limited number of pins is to reuse certain pins through

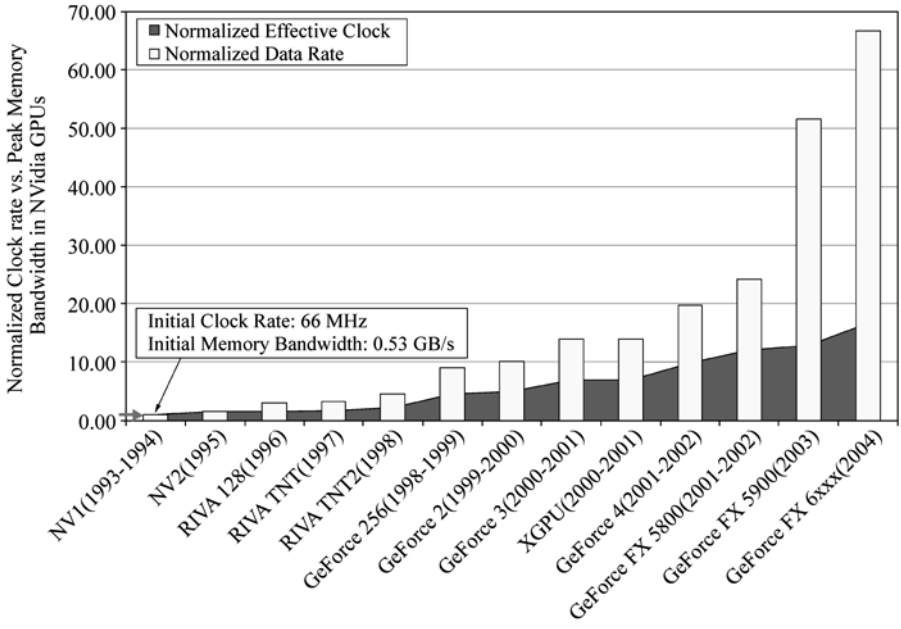


Figure 7.5 Normalized clock rate vs. peak memory bandwidth of NVidia

multiplexing at the cost of performance penalty. Another problem of the I/O pins is that they will be running at a very high clock frequency and thus consuming considerable amount of power. The above observation, although focused on GPUs, also applies to CPU-based applications like general-purpose computing, networking processing, and embedded systems.

In Chapter 3, we have investigated the potential of vertically stacking DRAM on top of a microprocessor to improve instruction throughput. Such a potential has important implications on the microprocessor industry and other bandwidth-demanding applications such as graphic processors and network processors. Meanwhile, it is very likely that the investment in developing 2.5-D integration technologies could be amortized by the large production volume of such products. Therefore, we envision that, therefore, in the timeline of three to five years the next major driver for the 2.5-D integration would the stacking of logic and memory chips for memory bandwidth demanding applications.

Future general purpose processor and application specific processor will inevitably take the path of parallel processing because of the mismatch of device and wire delays (e.g., the area percentage of a microprocessor that can be reached in one clock cycle will be less than 0.4% at the 35nm node[9]). Accordingly, future processors are likely to be organized as a tile-based multi-processor architecture with a 2-D array of regularly placed tiles[11–13]. A tile can be composed of a CPU, an application specific accelerator, or a DRAM block. These tiles work concurrently and communicate with each other through traditional wires or by sending packets.

It turns out that the 2.5-D regime is an ideal solution for a tiled multiprocessing architecture. The logic tile can be mapped to a high-performance CMOS chip and DRAM tiles can be allocated to a dedicated DRAM chip. The interface between logic and DRAM can be through inter-chip interconnects. With memory chips

directly attached on top of logic chips, inter-chip contacts could provide more than enough interconnects for the data traffic between the processing and memory tiles. The logic-memory interface can be implemented as low voltage logic running at a relatively low clock frequency as long as there are enough pins. Thus, significant power saving is expected because the power consumption is linearly proportional to the square of voltages. In addition, since memory interface (e.g., access protocol, location of inter-chip contacts, etc.) can be standardized and both CPU and memory chips are fabricated in large volume, the investment in developing stack technologies can be amortized by a large product volume. Such a system is shown in Fig. 7.6.

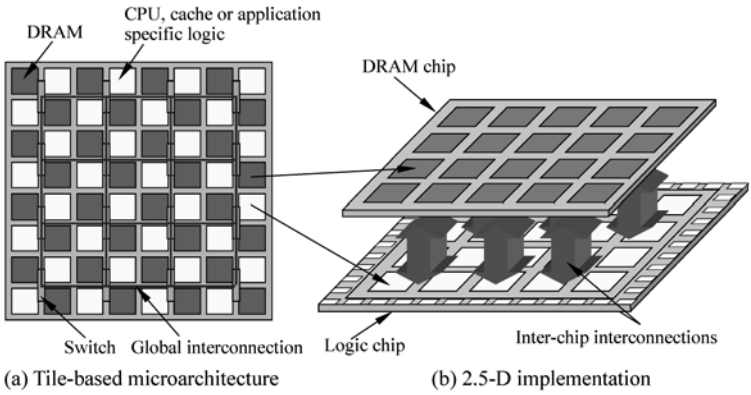


Figure 7.6 Tile-based multiprocessor architecture

7.3 Hybrid System Integration

The 2.5-D integration scheme provides a natural solution for future wireless chipsets where hybrid technology parts have to be integrated. We envisage that the next milestone of 2.5-D integration would be its deployment in the wireless terminals.

The communication operation of cellular phones depends on the synergy of a set of independent chips fabricated with different technologies. Today a typical wireless solution contains at least 6 different technologies: Bi-CMOS for the radio transceiver, analog CMOS for the radio and audio codec, digital CMOS for digital baseband processor and application processor, flash memory, high voltage CMOS for power management, and passives such as SAW filters and inductors[14]. Figure 7.7 illustrates Texas Instruments' multi-chip platform for a UMTS handset[15].

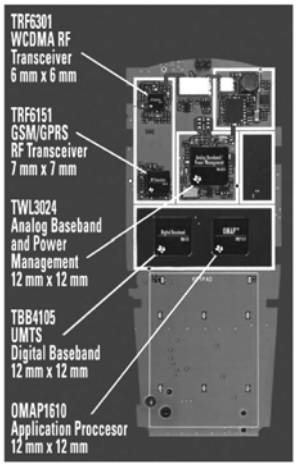


Figure 7.7 A multi-chip wireless handset solution[15] (courtesy of Texas Instruments)

An accelerating trend for today's cell phones is to support multi-mode, multi-standard, and multi-band operations, which require even more complex RF front-end circuits. Meanwhile, the communication-oriented cell phone standards are converging with wireless broadband access standards WiFi[16] and WiMax (806.16d/e)[17]. Wireless terminals are expected to be able to deliver computing, computation, and entertainment functionalities virtually anytime, anywhere. As a result, a large variety of heterogeneous functionalities could be found on the

handset: video/audio codec, graphics, camera, BlueTooth, GPS, DRAM, Flash storage/memory card, FM radio, MEMS accelerometer, USB, and so on.

To deliver the above functionalities with stringent size, weight, and battery life constraints, the integration capacity of the wireless chipset has to continuously increase. Although it is likely that designers would likely to consolidate all digital components into one single chip in the next wave of integration, how to integrate the digital and analog components is still a widely disputed issue (e.g., [18−21]). To date, the 'single-chip' integrated wireless solution is hindered by the many factors as discussed.

Inability to Build High Quality Passive Components The quality of on-chip passive components, especially spiral inductors, is far from satisfactory for general wireless phone service. To our best knowledge, the quality (Q) factor of on-chip silicon inductors is only between 10−20[22], while off-chip inductors could easily achieve a Q factor of 50. Since the phase noise of VCO is primarily determined by the Q factor, fully integrated VCOs suffer from a limited tuning range ($<\pm 10\%$). As a result, a multi-band, multi-mode handset needs multiple VCOs, which occupy a large silicon area. On the other hand, if the tuning range could be made larger than $\pm 20\%$, a single VCO could cover up the multiple frequency bands of a multi-band, multi-mode handset[21].

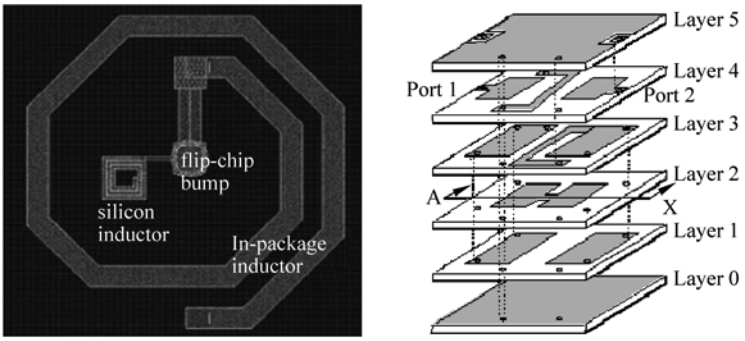
Process Incompatibility While digital CMOS technology will continue be the carrier of logic circuitry, RF CMOS is only proper for low-end applications with relatively low sensitivity requirements[22]. One major limitation of the CMOS process is the difficulty of isolation due to the relatively low resistivity of CMOS substrate, which is subject to excessively conductive loss from the standard of RF applications. On the other hand, a SiGe BiCMOS process has proven to be a very competitive candidate technology for RF components due to its superb RF

performance and ease to design with [23,24].

Scalability Issue Unlike their digital counterpart, analog circuits may not always benefit from technology scaling. For instance, some passive components have to occupy a relatively constant area to function properly. As a result, for a single-chip solution, the analog components will actually become more expensive when the fabrication process scales down. Meanwhile, a lower level of supply voltage would reduce the dynamic range of analog/RF circuits and impair the signal to noise ratio.

According to the above discussion, a single-chip solution for wireless handsets is unlikely to be able to deliver acceptable performance at a cost-efficient level. On the other hand, the 2.5-D scheme promises a very elegant way to integrate heterogeneous components fabricated with different technologies. As the first step of applying the new scheme to the mixed-signal domain, passive components can be embedded into the package or built on a separate substrate that could be stacked with other chips. Research prototypes (e.g., [25–27]) and early commercial products (e.g., [28,29]) have been reported. Figure 7.8 shows two example circuits (an in-package inductor and a passive filter) implemented in this manner. The advantage of this approach is impressive: the in-package passive components not only do not consume on-chip silicon real estate, but also could achieve a very high Q factor ($60 - 180$ at $1 - 3$ GHz[26]).

For wireless applications, the next milestone along the roadmap of the 2.5-D integration would be to build the system on different chips, each with the most proper technology optimized for performance and/or cost. The final system can be assembled by vertically stacking them. The inter-chip communication can be through inter-chip contacts built in between different chips. An extra advantage is 2.5-D integration naturally fits to a modular design style amenable to reuse. If



(a) In-package inductor (adapted from[26]) (b) Multi-layer filter (adapted from[26])

Figure 7.8 Passive components in package

each die is designed as a standardized module, a given multi-standard, multi-mode cellular chip-set can be implemented by stacking a corresponding combination of such dies. The add-on applications such as the digital camera, GPS and BlueTooth transceivers can be easily implemented as optional dies and only assembled regarding to user customization.

7.4 Extremely High Performance Systems

In the time frame of 10 years, we believe the 2.5-D integration will enable the VLSI systems to achieve superior performance that is impossible for their monolithic implementations. In this section we discuss two such examples.

In the time frame of 10 years, we believe the 2.5-D integration will enable the VLSI systems to achieve superior performance that are impossible for their monolithic implementations. In this section we discuss two such examples.

7.4.1 Highly Integrated Image Sensor System

As the first example of future high performance 2.5-D integrated systems, we

consider a focal plane system (sensitive to either visible lights or infrared radiation (e.g., [30,31]) deployed in the terminal-guidance module of air-combat missiles. As shown in Fig. 7.9, a typical focal consists of both digital and analog sub-systems. The analog sub-system is composed of a detector array and its readout circuits, i.e., analog signal processor (ASP), analog-digital conversion (ADC), and other auxiliary circuitry. The digital signals from the ADC are sent to the digital sub-system including digital signal processing (DSP), application processor, and memories. In current solutions, the analog sub-system can be built on a single chip[30], while the digital sub-system may need one or two logic chips as well as a set of memory chips.

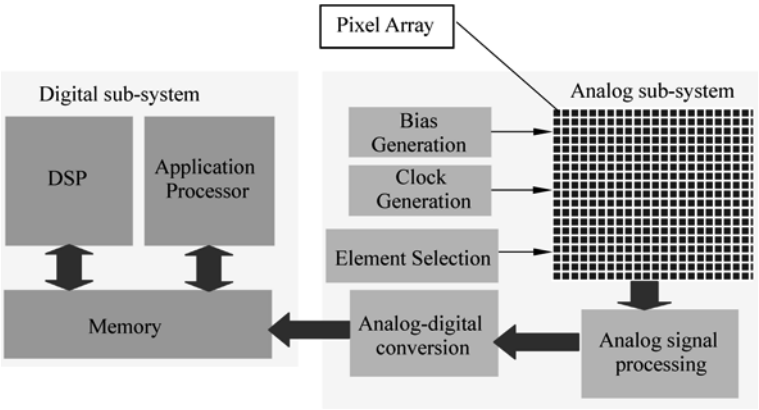


Figure 7.9 An image sensor system digram

The performance of current systems is restricted by several factors. First of all, high-quality detector depends on complex materials and processes, which are very difficult and/or costly to be integrated in a convention semiconductor process. Secondly, the inherent paralleling processing potential is poorly utilized. For example, although the photon detector cells are organized as a 2-D array, each column of

cells have to share the same ASPs and ADCs circuitry due to the space limitation. The problem is that the layouts of column ASP and ADC have to be designed to be very tall and thin to match the pitch of the detector cell. As a result, the noise in the output current due to spatial non-uniformity is extremely hard to overcome in such a configuration. Meanwhile, the bus interface between memories and logics may also become a performance bottleneck. Thirdly, the ADC must run at a very high clock frequency so that all cells in one column can be processed in a timed manner. However, a high-speed ADC demands large power consumption. Finally, due to the limit of pixel pitch and chip area, designers have to choose long signal wires and/or a simplified circuit structure, which could further impair the scalability and performance of IR system.

Under the 2.5-D integration regime, the image sensor can be integrated in a 3-D 'cube' as shown in Fig. 7.10. The whole system consists of a stack of chips, which are built in different technologies. The first slice in the cube is the detector array, which can be implemented in a specific process to optimize sensitivity to the input photonic signals. Note that the substrate of detector materials is usually transparent. Hence, the detector chip can be face-to-face bonded with the analog signal processing circuitry built on the second slice to maximize sampling efficiency. The third slice from the top is an ADC layer. Since a pixel in the detector has an area of ~15 μm × 15 μm, both the ASP and ADC can be implemented at the pixel level[32,33] so as to fully utilize the inherent concurrency. This level of parallel processing would allow the ASP and ADC circuitry to work with a very low supply voltage for power saving. After analog-to-digital conversion, the digital image can be stored in the memory chip built in a dedicated DRAM process. Since one frame of image only requires one storing process, the ADC slice and memory slice can be bonded in a back-to-back manner and interconnected

through a centralized bus. The bottom slice in the cube contains all the digital logic fabricated in standard CMOS. The digital logics, especially performance-oriented DSP circuits, can be designed with multiple processing pipelines with separate memory interfaces to fully utilize the extra bandwidth by the inter-chip interconnects.

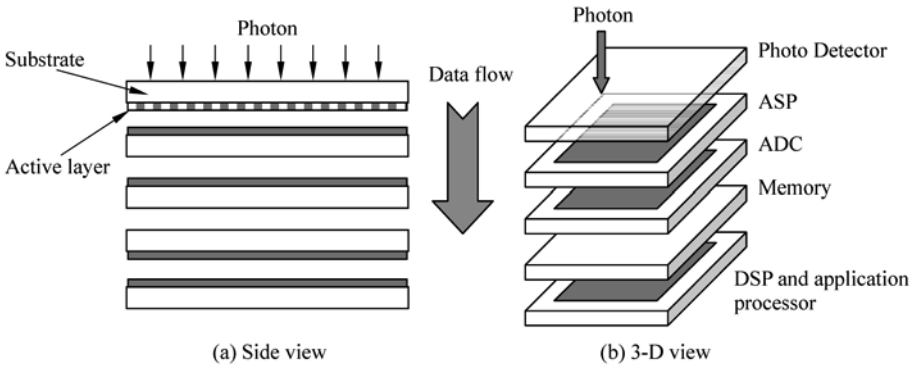


Figure 7.10 A 2.5-D camera/IR sensor system

7.4.2 Radar-in-Cube

As a second example, we consider high-performance, mission-specific radars deployed in advanced weapon systems such as unmanned aerial vehicles (UAVs), air-to-air combat missiles, and cruise missiles. Modern war environments have posed very stringent performance requirements for mission-specific radars. The required computation efficiency against volume and power consumption for different carriers is shown in Fig. 7.11. The problem is that, the inherent architectural complexity of radar systems (Figure 7.12 is a typical system diagram) is far beyond the capacity of monolithic integration, while it is very challenging for a multi-chip solution to meet all the requirements.

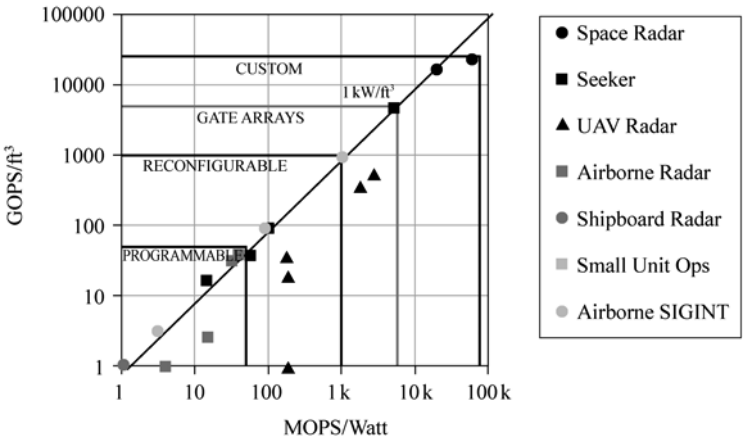


Figure 7.11 Computational demands for military radar systems (adapted from[34])

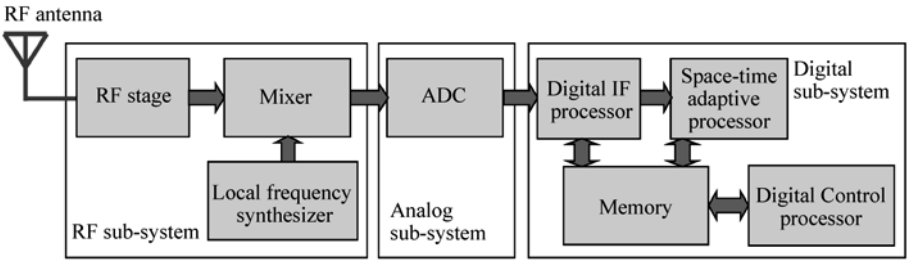


Figure 7.12 Block diagram of a radar system

It turns out that the 2.5-D "radar-in-cube" offers a shortcut path to build future high-performance military radars. Again the system can be integrated in a layered manner as illustrated in Fig. 7.13. The chips could be naturally organized into vertical pipeline stages for signal processing, while major system component could be separately tuned and manufactured for the highest performance. Meanwhile, if enough parallelism could be extracted, the whole system could work at a lowered voltage supply. Another advantage is the small size and weight due to the removal of board level package.

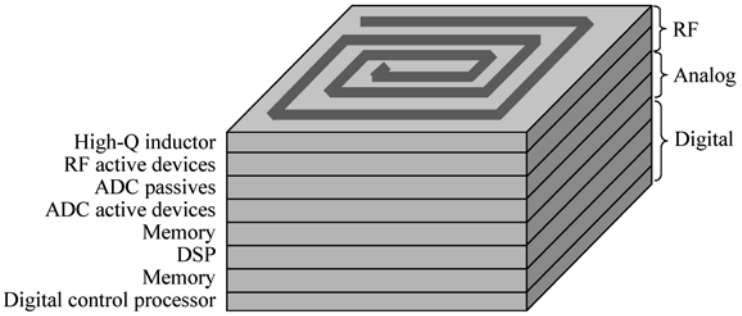


Figure 7.13 2.5-D implementation of a radar system

The design of a radar-in-cube would certainly be very challenging. Among various difficulties, the electromagnetic interferences between neighboring device layers must be contained. Accordingly, novel isolation techniques need to be developed to preclude electromagnetic noise. In addition, highly efficient electromagnetic filed solvers are crucial to provide full-system analysis so that the electromagnetic noise could be under control.

References

[1] Y. W. Heo, A. Yoshida, R. Groover. Advances in 3-D packaging—trends and technologies for multi-chip die and package stack. VLSI Packaging Workshop of Japan, 2002.

[2] M. Kada. The dawn of 3-D packaging a system-in-package (SIP). Sharp Electronics Corp. White Paper. [online]. Available: http://www.sharpsma.com/pub/productfocus/publications/memory/tec_whitepaper_dawn_of_3-D_pkgs.pdf.

[3] M. M. Santoyo. 3-D packing helps create small, powerful, and out of this world consumer devices. Electronics Journal, Nov./Dec. 2001, pp. 33 – 36.

[4] Intel Corporation. Intel stack Chip Scale Packaging products. Intel Flash Memory Home. [online]. Available: http://www.intel.com/design/flcomp/prodbref/298051.htm.

[5] G. Desoli, E. Filippi. An outlook on the evolution of mobile terminals: from monolithic to modular multi-radio, multi-application platforms. IEEE Circuits and Systems Magazine, 2nd Quarter, 2006, pp. 17 – 29.

[6] G. Purvis. Demands on 3G memory set to soar. Wirelessweb. [online]. Available: http://wireless.iop.org/articles/feature/2/6/2/1.

[7] M. Ieong, et al.. Three dimensional CMOS devices and integrated circuits. In: Proc. Custom Integrated Circuits Conf., 2003, pp 207 – 213.

[8] NVidia Corporation. [online]. Available: http://www.nvidia.com.

[9] Area3-D.net. Graphics chip table—reference. [online]. Available: http://www.area3-D.net/overview.php.

[10] Plasma online. identify NVidia chips. [online]. Available: http://www.plasma-online.de/index.html?content = http%3A//www.plasma-online.de/english/identify/picture/nvidia.html.

[11] M. Taylor, et al.. The Raw Microprocessor: A Computational Fabric for Software Circuits and General-Purpose Programs. IEEE Micro, March 2002, pp. 25 – 35.

[12] K. Mai, et al.. Smart memories: a modular reconfigurable architecture. In: Proc. The 27th Annual Int'l Symposium on Computer Architecture, 2000, pp. 161 – 171.

[13] R. Nagarajan, K. Sankaralingam, D. Burger, W. Keckler. A design space evaluation of grid processor architectures. In: Proc. The 34th Int'l Symposium on Microarchitecture, 2001, pp. 40 – 51.

[14] D. Buss, et al.. SOC CMOS technology for personal Internet products. IEEE Trans. On Electronics Devices, Vol. 50, No. 3, March 2003, pp. 546 – 556.

[15] Texas Instruments. TCS4105 UMTS Reference Design. [online]. Available: http://focus.ti.com/docs/apps/catalog/general/applications.jhtml?templateId = 1108&path = templatedata/cm/general/data/wire_chipset_umtsrefdesign.

[16] IEEE 802.11. LAN/MAN wireless LANS. [online]. Available: http://standards.ieee.org/getieee802/802.11.html.

[17] IEEE 802.16. LAN/MAN Broadband Wireless LANS. [online]. Available: http://standards.ieee.org/getieee802/802.16.html.

[18] K. Hansen. Wireless RF design challenges. In: Proc. IEEE Radio Frequency Integrated Circuits Symposium, 2003, pp. 3 – 7.

[19] DAC Panel. Mixed signals on mixed-signal: the right next technology. In: Proc. Design Automation Conf., 2003, pp. 278 – 279.

[20] A. Springer, L. Maurer, R. Weigel. RF system concepts for highly integrated RFICs for W-CDMA mobile radio terminals. IEEE Trans. On Microwave Theory and Techniques, Vol. 50, No. 1, Jan. 2002, pp. 254 – 267.

[21] L. E. Larson. Silicon technology tradeoffs for radio-frequency/mixed-signal system-on-a-chip. IEEE Trans. On Electron Devices, Vol. 50, No. 3, March 2003, pp. 683 – 699.

[22] J. Sevenhans, F. O. Eynde, P. Reusens. The silicon radio decade. IEEE Trans. On Microwave Theory and Techniques, Vol. 50, No. 1, Jan. 2002, pp. 235 – 244.

[23] J. D. Cressler. SiGe HBT technology: a new contender for Si-based RF and microwave circuit applications. IEEE Trans. On Microwave Theory and Techniques, Vol. 46, No. 5, May 1998, pp. 572 – 589.

[24] D. Y. C. Lie, et al.. RF-SoC: low-power single-chip design using Si/SiGe BiCMOS technology. In: Proc. Int'l Conf. on Microwave and Millimeter Wave Technology, 2002, pp. 30 – 37.

[25] K. Lim, et al.. RF-System-On-Package (SOP) for wireless communications. IEEE Microwave Magazine, March 2002, pp. 88 – 99.

[26] V. Sundaram, et al.. Digital and RF integration in System-on-a-Package (SOP). In: Proc. Electronic Components and Technology Conf., 2002, pp. 646 – 650.

[27] S. Donnay, et al.. Chip-package codesign of a low-power 5-GHz RF front end. Proc. of IEEE, Vol. 88, No. 10, Oct. 2000, pp. 1583 – 1597.

[28] R. J. Zavrel Jr., S. Bantas, S. A. Helic, R. Wood. Integration of silicon with passive devices yields advantages in wireless design. High Frequency Electronics, May 2003, pp. 56 – 61.

[29] D. J. Mathews, M. P. Gaynor. RF System in Package: Considerations, Technologies and Solutions. Chip Scale Review, Jul. 2003.

[30] C.-C. Hsieh, C.-Y. Wu, F.-W. Jih, T.-P. Sun. Focal-plane-arays and CMOS readout techniques of infrared imaging systems. IEEE Trans. Circuits and Systems for Video Technology, Vol. 7, No. 4, Aug. 1997, pp. 594 – 605.

[31] E. Fossum. CMOS image sensors: electronic camera-on-a-chip. IEEE Trans. On Electronic Devices, Oct. 1997, pp. 1689 – 1698.

[32] W. Mandl, J. Kennedy, M. Chu. MOSAD IR focal plane per pixel A/D development. SPIE Proceedings, Vol. 2745, Apr. 1996.

[33] J. J. Niewadomski, B. S. Carlson. CMOS read-out IC with op-amp pixel amplifier for infrared focal plane arrays. In: Proc. 10th IEEE Annual Int'l ASIC Conference and Exhibit, 1997, pp. 69 – 73.

[34] R. Reuss. Mission specific processing (MSP). DARPA TTO Program Review. [online]. Available: http://microsys6.engr.utk.edu/~bouldin/darpa/msp_public.pdf.

8 Conclusion and Future Work

Yangdong (Steve) Deng

Institute of Microelectronics, Tsinghua University
Beijing 100084, P. R. China, dengyd@tsinghua.edu.cn

Wojciech P. Maly

Electrical and Computer Engineering Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213-3891 USA, maly@ece.cmu.edu

Abstract In this chapter we conclude the book by reviewing the work we have done to validate the potential of the 2.5-D integration scheme. Moreover, we summarize the major challenges that need to be addressed by 2.5-D VLSI design, testing, and fabrication technologies and then propose tentative solutions.

Keywords 2.5-D Integration, cost analysis, design case study, 2.5-D physical design, fabrication, testing, design technology.

Today's consumers of electronic appliances have become used to each new generation of hardware running faster and offering more functionality but likely at a lower price. The incredible pace of development is enabled by the capability

of the underlying semiconductor industry to continuously deliver increasing integration capacity. The monolithic integration paradigm, however, is facing many challenges with the aggressive scaling down of feature size. Especially, it is inherently difficult and costly to handle long wires and mixed-technology components if the whole system is to be on a single chip. To address the above problems, while being able to maintain the momentum of IC functionality increase, this research focuses on investigating the 2.5-D circuit integration paradigm. In the last chapter of this book, we conclude major contributions of this work and propose future research directions.

8.1 Main Contributions and Conclusions

Through this work, we performed a series of research projects to study the feasibility of 2.5-D integration from both cost and performance viewpoints. The main contributions are summarized below.

A Cost Analysis Framework We constructed a cost analysis framework to compare various VLSI integration strategies on a unified basis. The cost is measured in terms of the actual silicon area consumed to build a working VLSI system. By translating the testing cost into an equivalent silicon area, our cost model could take into account integration styles involving components with varying defect coverage levels. The monolithic System-on-Chip and four non-monolithic integration schemes, Multiple-Reticle Wafer, Multi-Chip Module, 2.5-D integration and 3-D integration, are considered for the implementation of a target VLSI application with a working silicon area of 4 cm^2. It is assamed that the target application is assumed can be partitioned into multiple parts with identical area and all the

parts can be manufactured with the same CMOS process. With our cost analysis framework, it has been shown that the 2.5-D integration approach is noticeably more cost-effective than other integration schemes under a set of conditions that can be met through proper extension of today's technologies. In fact, the 2.5-D integration paradigm could be less expensive than the monolithic integration approach by more than 60%.

Design Case Studies Using the 2.5-D Integration Scheme A series of custom design case studies were conducted to evaluate the potential of the 2.5-D integration schemes. Specifically, we compared the monolithic and 2.5-D integration schemes regarding three different metrics: geometrical characteristics, timing performance, and system level throughput. The experimental results proved that the 2.5-D integration strategy offers superior flexibility in layout efficiency. It is generally possible to find efficient ways to 'fold' the long signal path in a 2.5-D implementation so that long wire length could be avoided. Meanwhile, the integration scheme provides a natural way to combine different technologies and design paradigms in a 2.5-D integration system. One such example is that a microprocessor and DRAM chips can be individually manufactured and then stacked together without the need for off-chip memory bus. A properly designed 2.5-D VLSI system could thus potentially achieve higher performance in terms of timing delay and system level instruction throughput.

A Series of 2.5-D Physical Design Tools Due to the complexity of modern ASIC designs, automatic EDA tools are critical for a successful layout implementation. For 2.5-D integrated ASIC systems, additional complexity is introduced by the large amount of inter-chip communication resource. To be able to pack an ASIC system in a 2.5-D space, we developed 2.5-D floorplanning, placement and

routing tools. We also realized that different design scenarios have to be distinguished for various design styles. The first scenario is for high granularity component based designs. For these designs, we apply floorplanning tools to manipulate functional blocks with arbitrary rectilinear shapes. Given the 2.5D integration paradigm, we need to further differentiate two situations. Under the first situation, all the blocks are undividable, which means although the netlist can be assigned to two chips in a 2.5-D system, each block can only stay in a specific chip. Under the second situation, some blocks would be 2.5-D dividable, which means that they can be split into two chips. The second ASIC design scenario is low granularity component based designs. Here we use cell placement tools to deal with standard cells. Yet another ASIC design scenario is mixed granularity designs, where we need to use mixed placement tools to handle both macros and cells. Our tools can be organized into diverse flows for the need of different design styles targeting a 2.5-D layout space.

2.5-D Layout Tools Enabled Design Case Studies We performed a large number of ASIC designs using our 2.5-D physical design tools. These designs include both academic benchmark circuits and industry applications with varying functionalities and complexities. For all ASIC designs, we evaluated the feasibility of the 2.5-D integration by comparing the interconnection characteristics between the monolithic and 2.5-D layout implementations. The results show that the 2.5-D integration has a potential for achieving speed/power performances superior to equivalent monolithic SoCs. Our experiments also reveal that the potential of the 2.5-D integration could only be fully unleashed only if proper automatic design tools are available.

8.2 Future Work

With the results from the work reported in this book, we can now answer many of the questions that did not have clear answers earlier. Moreover, this research opens the path for many new research directions. Since we have already justified the superiority of the 2.5-D integration from cost and performance perspectives, future work should focus on developing fabrication and test technologies to deliver the advantages, as well as efficient system architectures and design tools to fully utilize the potential.

In this section, we will outline important research to address the design, testing, and manufacturing issues for 2.5-D ICs illustrated in the Fig. 8.1. below. In Fig. 8.1, the two chips are face to face bonded to ensure short vertical inter-chip connections. A heat sink is attached on the bulk side of one chip, while flip-chip based area I/Os are deployed on the back of the other chip. High performance circuits should be

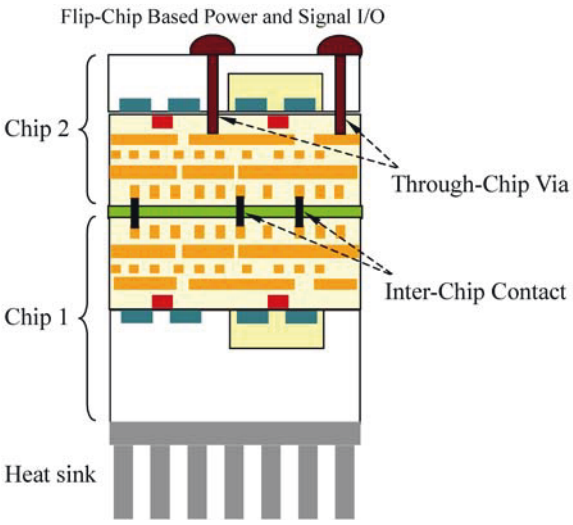


Figure 8.1 Area power I/O for 2.5-D integration (*see colour plate*)

assigned to the chip with its bulk attached to the heat sink so that the heat build-up can be better controlled. Both inter-chip contacts and through-chip vias need to be deployed for different types of interconnects. In addition, the upper chip has to be thinned to accommodate the through-chip vias so that they can be built with reasonable height and pitch.

8.2.1 Fabrication Technology for 2.5-D Systems

To achieve true 2.5-D integration, new fabrication technologies have to be established. In fact, different chips in a 2.5-D integrated system can be separately manufactured with a conventional technology. The most challenging part with regcard to the process side is how to build the inter-chip interconnections. In the first chapter, we already briefly reviewed the available technologies to interconnect two layers of chips under the context of 3-D integration. With these technologies, however, the yield loss has to accumulate during the extra processing steps. As discussed in Chapter 2, a 3-D IC would be more expensive than its single-chip version. To avoid the cumulative yield loss, we believe that the 2.5-D integration has to depend on a MEMS based assembling/dissembling technology.

The key idea is to build a MEMS-based mechanical latch as the interconnection gadget. The MEMS-based technology will center on the concept of laterally compliant contacts for chip-to-chip interconnection. Figure 8.2 shows a schematic of the laterally compliant contact with a clamp extending from its free end. For the purpose of this discussion, for now we would assume that the latch is on the bottom chip. The corresponding interconnect on the top chip is built as a cylindrical stub and can be latched by the clamp. The cantilever-clamp offers an attractive solution to tradeoff between compliance and contact forces. Sufficient compliance

is desired for compensation of imprecision that arises during fabrication and alignment processes as well as thermal mismatch effects, while the contact must be stiff enough to ensure an acceptable contact force. The cantilever-clamp decouples these two conflicting design needs, as the cantilever can be highly compliant while the clamp can be stiff. With current MEMS technology, it is estimated that our laterally compliant contacts will have a footprint smaller than 20 μm × 5 μm[1].

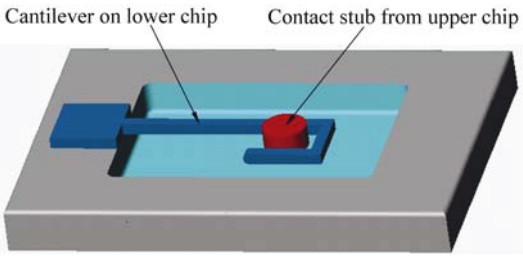


Figure 8.2 MEMS based inter-chip contact (*see colour plate*)

To assemble two chips, a sliding force is applied to either the top chip or the bottom chip, and thus the posts that protrude from the upper chip can be guided into recesses etched into the lower chip. As each post is pushed from the wide end to the narrow end of the corresponding recess, alignment is passively refined[1]. The whole process can be organized into two succeeding stages depending on alignment features of varying sizes so that the accuracy can be successively refined. The first stage alignment posts are relatively thick for rough alignment and could be fabricated from additional thin film deposition and patterning on a CMOS chip. On the other hand, the second stage alignment posts can be fabricated from the same metallization layers on the CMOS chip as the inter-chip contacts, and are therefore highly precise. Upon the completion of the second stage alignment, the contact elements can be precisely engaged to form the contact. Such a passive alignment

methodology is critical for precise engagement of small footprint lateral contacts.

During the assembling process, a faulty die could be ‘plugged’ and thus lead to accumulative yield loss. To counteract such yield loss, we envision the MEMS latch could support re-work to a certain extent. For instance, when a proper force in the opposite direction to the assembling process is exerted, the two assembled chips can be dissembled. The abilities of assemble and dissemble are critical, because otherwise it would be too expensive to build a 2.5-D system.

When more than two layers of chips are to be stacked in a 2.5-D system, all three possible combinations of bonding styles, face-to-face, face-to-back, and back-to-back, might have to be applied. To provide a complete interconnection solution in the vertical direction through the stack of dies, the laterally compliant contacts must be supplemented by through-chip interconnects. Such technologies have been developed under the context of both 3-D integration and MEMS. By fine tuning processes in high aspect ratio deep reactive ion etching (DRIE), chip thinning, and low temperature dielectric deposition and electroplating (please refer to survey papers, e.g., [2]), it is feasible to fabricate through-chip interconnection holes down to a few microns in diameter.

8.2.2 Testing Techniques for 2.5-D Integration

Acceptable yield is a key requirement for the success of 2.5-D IC technologies. Effective yield of a stacked IC system will be no better than a very large monolithic implementation unless it is possible to test each device layer during the production and possibly correcting or skipping the layer. Testability during manufacturing is a key problem and requires careful planning of controllability and observability points on a layer by layer basis.

In the first chapter, we briefly reviewed the potential techniques to solve the testing problem for 2.5-D integrated systems. Based on various isolation and self-testing methodologies, these testing solutions could well handle the block-based design style in which every functional block is located within a specific chip and the inter-chip interconnections are only assigned to inter-block wires.

If the logic assigned to a given layer chip is not self-contained, however, the above testing solutions can not be directly applied. As mentioned before, a random logic-based function block can be split into two chips if we take a cell-based design style and then perform a 2.5-D placement. The resultant number of interconnections between two chips could be significant. In fact, we can have as many inter-chip contacts as the bonding technology allows, as long as the introduction of them could improve system performance. In our experiments, we have seen a 100 K-gate circuit partitioned into two parts with ~10 K inter-chip contacts between the two parts. When the first chip in a 2.5-D IC is fabricated, however, the resultant functional block will only have a partial netlist available and it would be extremely difficult to test it before the upper-layer chip is stacked. There are two difficulties in this regard, 1) inter-chip contacts are too small for tester access, and 2) the number of inter-chip contacts is far beyond of the capacity of current testers. The partition also introduces problems for scan-based Design-For-Test (DFT) techniques when the cells on a scan chain are assigned into two chips.

We believe that 2.5-D DFT techniques should be developed to resolve the above problems. The test data compression technique would be essential to test a partial netlist with a large number of inter-chip contacts based I/O. Extra testing circuitry should be inserted so that compressed test results can be accessed through conventional testing pads (e.g., on the boundary) of a chip. Meanwhile, scan chains

have to be designed to fully connect the inter-chip contacts. In other words, each chip has to be equipped with one or more scan chains so as to guarantee the controllability and observability of both registers and inter-chip contacts. Only with both the test compression and scan chain techniques could a 2.5-D IC be tested in a hierarchical and incremental manner.

Fortunately, as shown in Chapter 2, 2.5-D integration can be more cost effective than other approaches in a wide range of defect coverage. As long as we could limit the untested chip area to a certain level, the 2.5-D solution could still outperform other approaches from a cost point of view. Here a key implication is that the testing requirements must be honored by 2.5-D physical design tools so as not to generate a partition solution with too low a defect coverage level.

8.2.3 Design Technology for 2.5-D Integration

Given the option of 2.5-D integration, the design of future VLSI systems is likely to be further complicated. Novel EDA algorithms and tools have to be developed to address design challenges in the following three categories.

Complexity Effects Essential factors leading to the complexity issue include the choice of process technology (monolithic versus 3-D, low leakage vs. high performance CMOS, RF CMOS vs. GaAs), the selection of power distribution network (power mesh, area power I/O vs. peripheral power pads), the combination of design technologies (digital, analog, optical, MEMS), the necessity for reusing verified components (IP cores, standard buses, memory), the huge number of transistors for both logic and memory devices, and the wide range of communication protocols available for inter-core transactions (buses, on-chip networks, asynchronous, globally asynchronous locally synchronous).

Thermal Effects Contributing factors to the heat problem are power density, material properties in terms of heat conductivity, placement and performance of the heat sinks and other heat dissipation features, Joule heating effects, non-uniform substrate temperatures and thermal gradients.

Manufacturability Effects Critical factors resulting in manufacturability problem are physical phenomena that cause IC manufacturing failures, low yields due to defects and/or process variations, failure due to inter-layer connections for 2.5-D ICs, and signal integrity problems due to inaccurate models of the logic devices and interconnect.

To deal with these issues, 2.5-D IC designers must employ realistic models of the process technology, circuit fabrics and technology-aware design flows and tools. At the front end, exploration tools need to be developed to answer the what-if questions and serve the system advisor. Meanwhile, with the wide application of System-on-Chip in which one or (increasingly) multiple processors have to present in a system, system level EDA tools have to be built to optimize system architecture under the 2.5-D integration paradigm. At the back end, physical design tools have to pack an input netlist in a 2.5-D layout space. The tools constructed in this research could serve as the prototype for the development of future academic/commercial tool suite. In this sub-section, we outline the important EDA tools that need to be delivered in the coming years for the successful deployment of 2.5-D ICs.

8.2.3.1 2.5-D Architecture Exploration tools

The results reported in the previous chapters demonstrate that the 2.5-D architecture significantly reduces the total wire length and the longest individual

wire-length. The objective of this project is to identify and quantify tradeoffs related to complexity, thermal and manufacturability effects of 2.5D IC technologies and thereby enable an SoC designer to select the optimal 2.5-D technology for his/her design specification and requirements. Typical questions are how many silicon layers stacked on top of each other, how many metal interconnect layers per silicon substrate layers, how many access points to the outputs of logic cells, types of logic cells and level of supply voltage used on different silicon substrate layers, and so on. These issues can be addressed through development of analytical and empirical models for processes and devices and a prototype computer tool enabling tradeoff exploration in this complex space with emphasis on design yield.

Modeling Infrastructure To ensure the effectiveness of abstraction-based design methods for tradeoff exploration, accurate modeling of nanometer-scale effects is needed to allow meaningful evaluation of relevant design metrics at the higher levels. The models must cover not only individual device and wire characteristics, but also their behavior when interacting with other components to form circuits and gates and so on up the hierarchy. Power, performance, reliability and other metrics must be accurately modeled to enable synthesis, mapping and physical design algorithms to implement designs of minimal cost and/or highest performance.

Thermal Modeling A key stumbling block in the roadmap of 2.5-D integration is excessive heat generation in the 2.5-D stack and the rather limited ability for heat removal. Thus analysis tools have to be developed to precisely quantify the thermal effects in 2.5-D ICs. With these tools, designers could derive a temperature profile at different abstractions levels. The thermal distribution information would enable calculation of the mean time to failure (MTTF) and the self-heating effect

of interconnects. Another complexity arises from the fact that the leakage current is a strong function of temperature and it in turn could affect the thermal profile. Therefore, a coupled analysis/simulation environment is, therefore, essential to accurately characterize the thermal behavior of 2.5-D ICs.

Electromagnetic Interference (EMI) Modeling One advantage of 2.5-D integration is its viability to integrate chips fabricated in different technologies. However, when building 2.5-D integrated wireless chipset, however, the digital chips might be placed very close to RF circuits and thus the digital switching noise could seriously interfere with the RF operation[3]. As a result, the electromagnetic interference between digital and analog circuits must be properly modeled so that designers could identify major EMI sources/victims and employ various isolation features accordingly.

Power Distribution Modeling Today typically a considerable portion of chip area will be devoted to an on-chip power supply network consisting of power grid and decoupling capacitance. In a 2.5-D IC, the high-quality on-chip power supply network must be designed with exceptional care so as to use the chip area efficiently, eliminate potential electromigration failures, and avoid excessive voltage drops. Especially, both static IR-drop noise and dynamic Ldi/dt noise have to be well controlled by fine tuning the wire width of the power grid. Another extra complexity arises due to the uncertainty of workload because modern consumer devices tend to use a number of different power modes.

Parameter Variation Modeling The performance of future 2.5-D ICs will be increasingly affected by parameter variations. The sources of variations can be classified into two categories: 1) process variations due to the deviation of fabrication process, and 2) environment variations resulted from the change of

working conditions (e.g., temperature, supply voltage)[4,5]. The impact of parameter variations on different design metrics have to be accurately modeled so that statistical optimization techniques could be performed to derive a robust solution.

Automatic Exploration Tools With the above modeling infrastructure, an exploration engine can be employed find a (generally Pareto) optimal solution. The exploration engine has two major parts: an evaluation engine assessing the current solution, and an optimization engine generating the next solution.

Evaluation Engine The exploration operations have to be efficient enough to fast navigate a large solution space. At the present time, SystemC[6] provides an efficient compiled simulation framework for fast microarchitecture level evaluation. The exploration engine for 2.5-D ICs could be built on the basis of SystemC models. The simulation could derive importation like system level throughput and data traffic among system modules. The data traffic information could then be fed to a communication-driven 2.5-D floorplanner (instead of wire length driven because the connection information may not be well defined at this abstraction level)[7]. With the floorplan-level layout information provided, various analysis engines could use the system profiling information to assess the quality of current design solution.

Optimization Engine The optimization engine can be built in generic optimization frameworks including branch-and-bound, simulated annealing, and genetic algorithms. Since multiple optimization objectives have to be considered, the objective is to find either a Pareto optimal front or optimize a weighted sum of multiple object functions. Efficient heuristics would be required to prune the solution space and speed-up the optimization process.

8.2.3.2 System Level Design Tools

Future VLSI systems implemented in a 2.5-D integrated manner are likely to integrate one or more microprocessors backed up by a memory hierarchy. The system functionality will depend on the synergy of both hardware and software coordinated by an operating system. Recently system level design has attracted significant research effort (e.g., [8,9]). Under the 2.5-D integration paradigm, the current work must be extended to take into account a series of new issues.

Data Placement Given a processor in a 2.5-D multiprocessor system, different types of memory modules or memory banks belonging to one memory module may be built on different chips and thus have varying timing characteristics. Compilers and operation systems would have considerable freedom to control the heat build up in a working system. A central problem is how to map system and program data into different memory locations. Intuitively, one may consider performing a profiling to identify frequently accessed arrays and stack data and then map them to faster memory banks on "cooler" layers. Meanwhile, with the help of the operating system, it is possible to dynamically move data and map instruction address into memory modules distributed into multiple chips so as to reduce the heat dissipation in the 2.5-D structure. In case of dynamic data and code migration, we must account for the overhead of memory copy operation and contrast it with the power savings that may be achieved.

Dynamic Power Management and Dynamic Voltage and Frequency Scaling The key idea is to turn off or slow down the processing units in a 2.5-D IC when they are not used or are underutilized. For example, some of the processing circuitry in the 2.5-D IC may be power gated or clock gated when they are in deep sleep or standby, a processor could run in a lower-frequency mode when full processing

power is not necessary, and so on. A static approach is to assign the supply voltage level of different voltage islands in a given layer so that the expected temperature distribution could match the heat dissipation capability within each layer. Since high-speed global buses could be a significant source of heat build-up, the bus signaling mechanism (current signaling vs. voltage signaling, redundant vs. irredundant, etc.) and physical parameters must be carefully chosen so as to achieve a target performance goal while avoiding excessive heat generation and temperature-induced reliability problems. Using some error correction coding scheme appears to be necessary in 2.5-D designs because of the higher noise levels in the 2.5-D structures.

Architecture Selection The 2.5-D integration would enable revolutionary system architectures to exploit processing parallelism at various granularities (thread, task, instruction, etc.). Especially, new distributed system architectures will be made possible because processors could separately access memory blocks stacked on their top. With the extra parallelism extracted, a 2.5-D IC could deploy a large number of processing elements with lower working voltage and frequency so that power density could be lowered but still deliver target performance.

8.2.3.3 Physical Design Tool Suite for 2.5-D ASICs

In the work reported in this book, a prototyping layout synthesis framework for 2.5-D integrated VLSI systems has been constructed. To provide 2.5-D specific optimization, the existing tools have to be extended with new features and at the same time new tools have to be developed.

2.5-D Physical Design The physical design tools developed in this research could serve as the blueprint for future 2.5-D ASICs. Besides improving the

scalability and stability of tools, many new features have to be delivered to construct future 2.5-D IC design tools.

2.5-D Pin Assignment In a typical ASIC back-end design flow, I/O ports or pins of functional blocks have to be assigned to exact locations after creating the floorplan or coarse-level placement. For 2.5-D ICs, now a new kind of flexibility is that the pins can be located on either the boundary or the top of a block. Thus the 2.5-D pin assignment must be able to take advantage such flexibility. Simple heuristics for this purpose are building an obstacle-aware minimal-spanning tree or Steiner tree for each net. If congestion issues have to be accounted, a more expensive but more accurate approach is to call a 2.5-D global router.

Placement Migration for Routability Besides the conventional routability issues, the inter-chip wiring congestion can be caused by the mismatch between the routing demand for inter-chips contacts and the available routing resource. 2.5-D physical design tools can be enhanced with a routability driven migration process so that the demand for inter-chip contacts can be mitigated to an acceptable level by moving the least number of place-able objects.

Placement Migration for Timing Optimization In a 2.5-D IC, we may have more freedom to perform timing optimization. For instance, it would be very useful if we can tweak an existing placement solution for better timing by folding a long timing path. Alternatively, for a long wire in a given chip, we can drop one or more buffers on the other chip to reduce the wire delay.

Thermal Driven Layout Design Heat removal in 2.5-D ICs is likely to exacerbate and pose a greater challenge in thermal management. Meanwhile, control of leakage power and temperature-induced timing and reliability issues must be addressed in realization of such systems. Temperature-aware physical

design tools for 2.5-D ICs should be designed in such a way that the important interdependency between temperature and leakage power can be taken into account.

Thermal Driven Floorplan Design By developing efficient physical design optimization algorithms in conjunction with effective full-system thermal and leakage modeling, the 2.5-D floorplan design must be optimized by considering area, wire length, leakage, and temperature gradient. It should be noted that a complete run of coupled thermal-electro simulation could be too time-consuming to be embedded into the evaluation engine of a floorplanning tool. An efficient solution thus has to depend on interpolating pre-characterized data.

Placement Migration for Hot-Spot Removal The effective and effi-cient removal of heat produced by the active devices is essential to guarantee the correct functioning of a 2.5-D IC. It is essential to avoid putting too many active devices in a region where heat dissipation is difficult. We propose to develop a tool for placement migra-tion to achieve desirable heat map and to minimize the interconnect length or timing penalty.

Placement of Heat Removal Features The above techniques could improve the heat dissipation characteristics of 2.5-D ICs to a given extent. They could not, however, guarantee a satisfying temperature distribution under any conditions. Then it is necessary to place additional heat removal features including dummy vias (as heat pipes) and thermoelectric refrigerators[10]. The process can be guided by an incremental thermal analysis engine. The difficulty lies in the fact that there may not be enough free space near the hot spots to place the heat removal features. Accordingly, the problem can only be resolved at the floorplan design stage by reserving space to honor the thermal budget.

Power Distribution Power delivery in 2.5-D integrated circuits is one of

the most critical challenges influencing the overall system functionality and performance. The objective is to deliver one or more stable supply voltages with nominal variations to devices on different layers. The power supplies have to be distributed through inter-chip contacts with acceptable IR-drop levels. An extra level of complexity is that the dynamic Ldi/dt noise in one chip could affect the circuit operation in the higher and lower layers of chips.

2.5-D Power Grid Sizing and Decoupling Capacitance Insertion In a 2.5-D IC, a power grid structure has to be designed with the max-imal allowable voltage drop and current density on each device layer specified by the designer[11]. The power grid usually has a regular topology but the wire width, the number and positions of vias, and the positions of decoupling capacitors must be optimized such that the total power grid area and the incurred via costs are minimized. For 2.5-D ICs, it is likely to have several power grids for different voltage domains. In addition, the power rails have to be routed through different chips and thus the usage of inter-chip contacts must also be optimized.

Power I/O Buffer Placement In current high performance VLSI designs, area power I/Os are increasingly being deployed to deliver high quality power supply superior to the peripheral power pads. Adopting the area I/O for 2.5-D systems is complicated by the fact that the face-to-face bonding of two chips is more likely to deliver higher performance. Therefor, we envision that a 2.5-D IC could be built in a way illustrated by Fig. 8.3. Now two chips are face-to-face bonded with a heat sink attached on the back of one chip and flip-chip based area I/Os built on the back of the other chip (By “back” we mean the bulk silicon side of a chip). The inter-chip contacts could be fabricated with a shorter ‘vertical’ wire length to guarantee faster connections. On the other hand, the area power and signal I/Os

are routed in through-chip vias, which tend to have a bigger pitch and thus smaller resistivity. However, the through-chip vias would consume a given percentage of silicon area and thus must be considered a limited resource. Automatic power planning tools have to be developed to manipulate the number and location of power I/O bumps so as to optimize the induced voltage drop.

Statistical Design Optimization Our existing layout synthesis framework could be enhanced with a statistical static timing analysis engine to account for process variation (especially inter-chip) in a systematic manner. For instance, the gate sizing and buffer insertion engines, design parameters (e.g., gate oxide thickness, channel length) should be treated as random variables so that a statistically robust solution can be derived.

8.2.3.4 2.5-D VLSI Design Flow

With the design tools proposed earlier in the sub-section, a complete design flow for 2.5-D ICs can be established as shown in Fig. 8.3. The input can be in either a system-level description or a Register-Transfer Level (RTL) description of the target design. The system level description could model both software and hardware in a unified manner and would allow fast algorithm exploration, while the RTL description has more details and permits more accurate analysis. Currently, the conversion from a system level representation to an RTL one is done by hand, but new tools (e.g., Cynthesizer by Forte Design Systems[12] are appearing in the EDA market to solve the problem automatically.

The whole design flow can be organized into two stages: (1) Exploration stage that navigates through the solution space and rapidly evaluates the design tradeoffs; and (2) Implementation stage that synthesizes the final layout for tape-out.

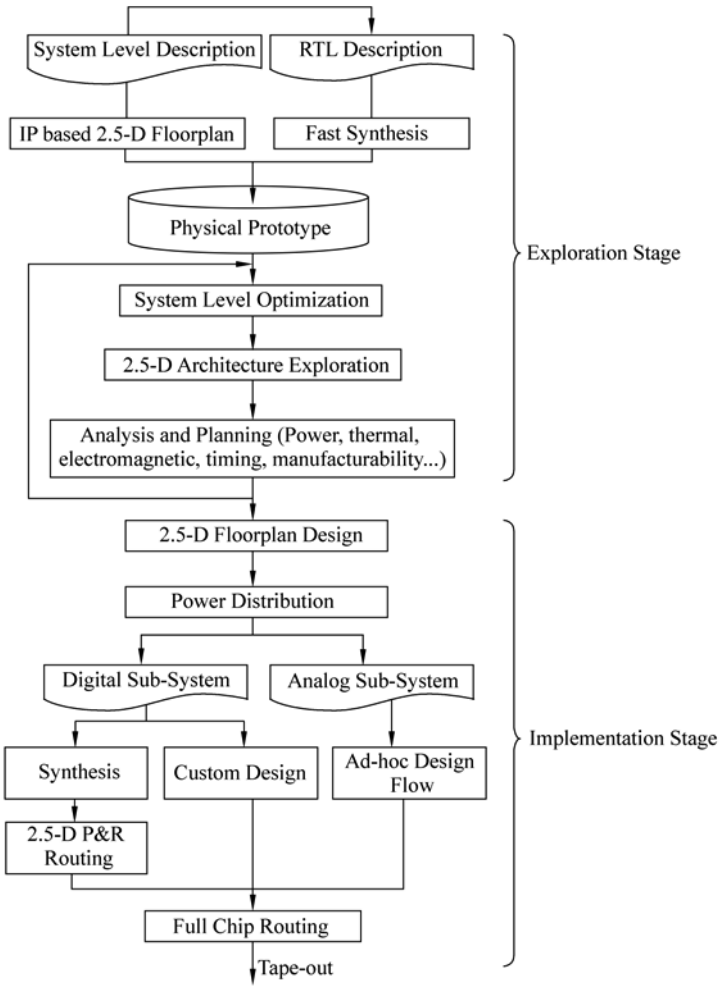


Figure 8.3 Design flow for 2.5-D ICs

Exploration Stage The system exploration stage is essential to answer the "what if" questions that arise in the design process. Starting from either a system level description or a RTL representation, a physical prototype (e.g., [13,14]) has to be built so that various physical effects can be evaluated. Given an RTL description, the physical prototype can be established through a fast synthesis followed by a

flattened, coarse-grained 2.5-D placement. To save CPU time, the placement process does not need to be completely finished as long as it can provide relatively accurate physical information. Based on the coarse-grained placement, a clustering process taking into account both geometrical closeness and logic hierarchy can be conducted to generate a design hierarchy that is proper for the actual RTL synthesis. Constructing a physical prototype from a system-level description can be achieved by either first translating it into an RTL representation, or directly performing 2.5-D floorplan design using physical information from IP characterization.

The physical prototype provides a platform for different analysis engines to extract various physical information including timing delay, power consumption, power supply characteristics, temperature profile, electromagnetic interference, and so on. The results can then be fed to a simulation engine to derive system level throughput, which could then be used by the exploration engine to optimize system configuration. In addition, compilers and operation systems could use the analysis results for both static and dynamic optimization for better performance and other metrics.

The output of the exploration stage is a system configuration in terms of the best number of wafers and metal layers as well as the optimum technology for each wafer. Meanwhile, the design objectives including timing, thermal, power distribution, electromagnetic, and other related budget will also be extracted.

Implementation Stage Starting from an optimized system configuration, the implementation stage could be adapted from today's advanced design flows (e.g., [13,14]). A hierarchical design style is required because different chips in a 2.5-D IC must be separately designed in parallel. Thus, the implementation design

stage should begin with a 2.5-D floorplan design to assign system components to different layers of chips. The floorplan solution could also allow designers to build a power grid structure.

Following the floorplan design, different sub-systems should be designed with proper flows, e.g., synthesis, placement and routing methodology for a general digital system, custom design style for datapaths and high-performance logics, and ad-hoc design methodologies for other specific functional blocks. The design of each block must honor the budgeting set in the previous design stage.

Next, all sub-systems should be integrated and incrementally optimized. The migration based technologies would be valuable because design iteration to an earlier stage would be costly in terms of design cost and turn-around time. Finally, the chip level routing could complete all necessary connections and make the system ready for final tape-out.

References

[1] Q. Lin. private communication.

[2] K. Banerjee, S. J. Souri, P. Kapur, K. C. Saraswat. 3-D ICs: a novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration. Proceedings of the IEEE, Vol. 89, 2001, pp. 602 – 633.

[3] T. Sudo, et al.. Electromagnetic Interference (EMI) of System-on-Package. IEEE Trans. on Advanced Packaging, Vol. 27, No. 2, May 2004, pp. 304 – 314.

[4] S. Borkar, et al.. Design and reliability challenges in nanometer technologies. In: Proc. Design Automation Conf., DAC 2004, pp. 75.

[5] S. Borkar, et al.. Parameter variations and impact on circuits and microarchitecture. In: Proc. Design Automation Conf., DAC 2003, pp. 338 – 342.

[6] SystemC community. http://www.systemc.org/.

[7] J. Hu, Y. Deng, R. Marculescu. System level point-to-point communication synthesis using floorplanning information. In: Proc. Asian and South Pacific-DAC/VLSI Conf., 2002, pp. 573 – 579.

[8] A. Jerraya, W. Wolf. Multiprocessor Systems-on-Chips. Morgan Kaufmann, Oct. 2004.

[9] B. Bailey, G. Martin, A. Piziali. ESL design and verification. Morgan Kaufmann Publishers, 2007.

[10] J. Sharp, J. Bierschenk, H. B. Lyon. Overview of solid-state thermoelectric refrigerators and possible applications to on-chip thermal management. Proceedings of the IEEE, Vol. 94, No. 8, Aug. 2006, pp. 1602 – 1612.

[11] H. Qian, S. R. Nassif, S. S. Sapatnekar. Power Grid Analysis using Random Walks. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 24, No. 8, Aug. 2005, pp. 1204 – 1224.

[12] C. Maxfield. Got system-level synthesis? [online]. Available. http://www.eetimes.com/news/design/showArticle.jhtml?articleID = 60402139. EE Times, 02/18/2005.

[13] Magma Design Automation. The FixedTiming® methodology. [online]. Available: http://www.magma-da.com/c/@saRbjwTRmvDx6/Pages/fixedtiming.html.

[14] Cadence. Encounter digital IC design platform. [online]. Available. http://www.cadence.com/products/digital_ic/index.aspx.

Index

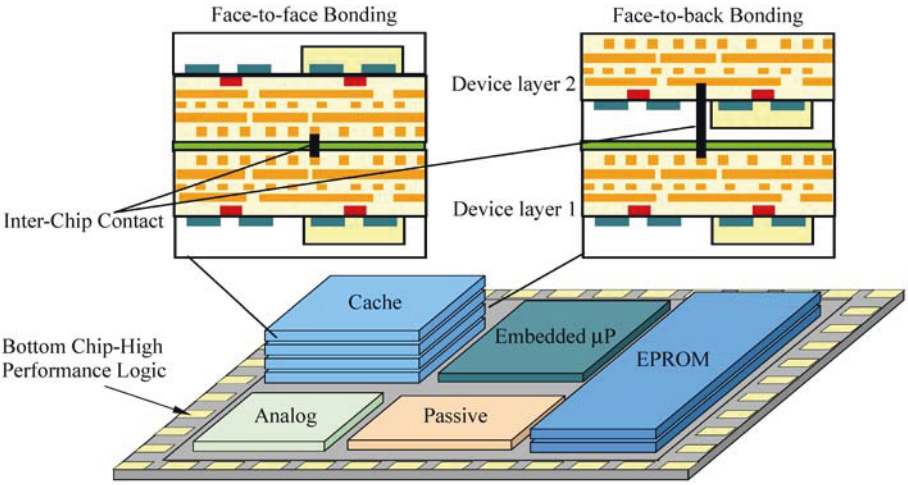


Figure 1.2 An imaginary 2.5-D system

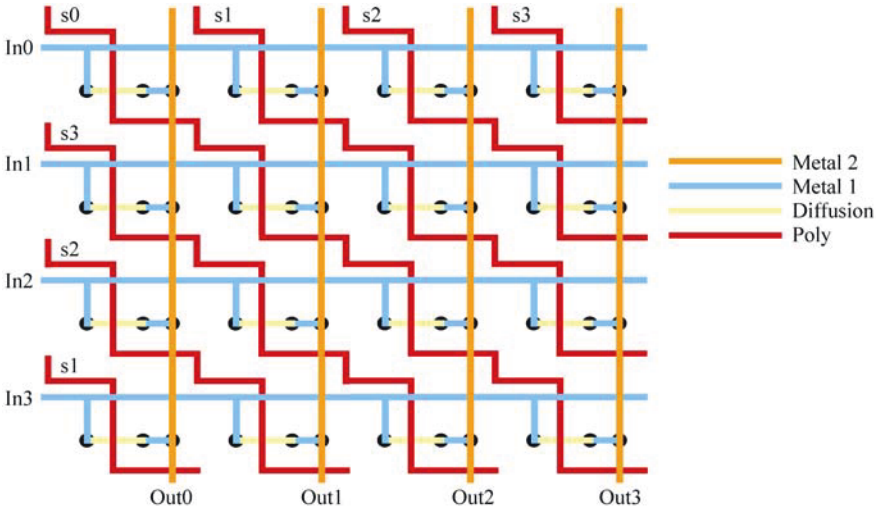


Figure 3.1 Stick diagram of a monolithic crossbar

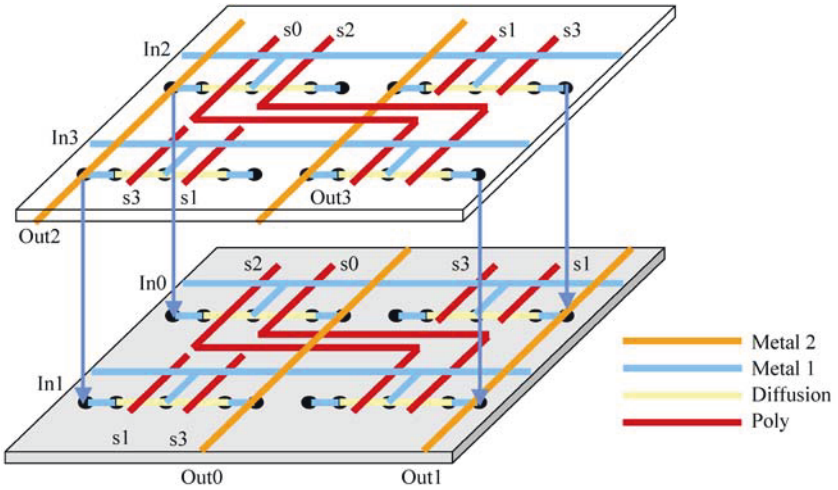


Figure 3.2 Stick diagram of a 2.5-D crossbar

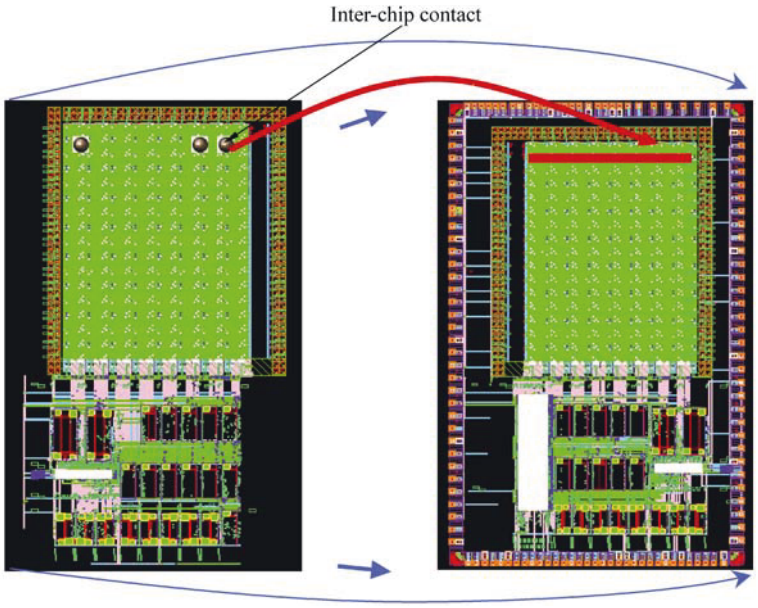


Figure 3.9 The 2.5-D re-design of PipeRench

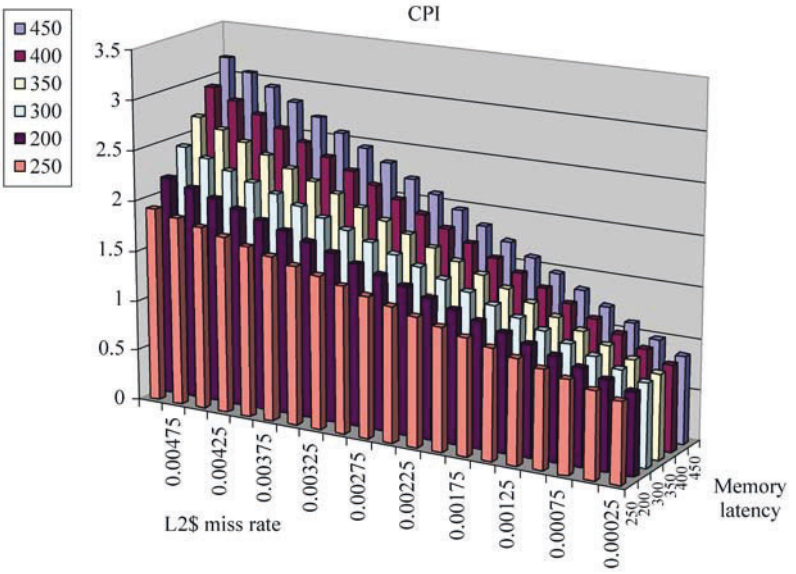


Figure 3.14 CPI with regard to main memory latency and L2 cache miss rate

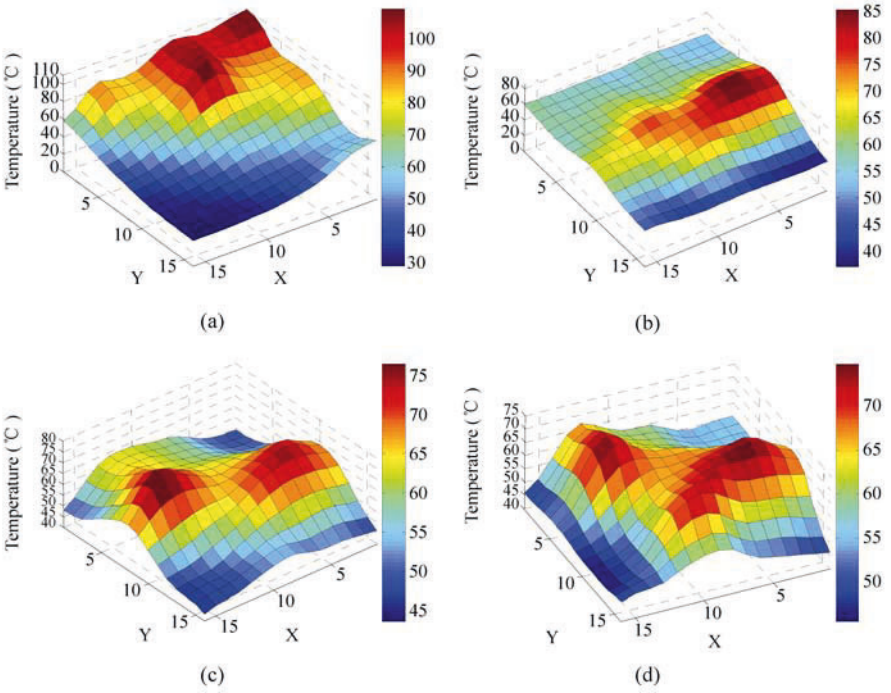


Figure 5.11 Temperature snapshots of the thermal driven floorplanning with Benchmark AMI49. Both the maximum temperature and the temperature gradient are reduced during the optimization

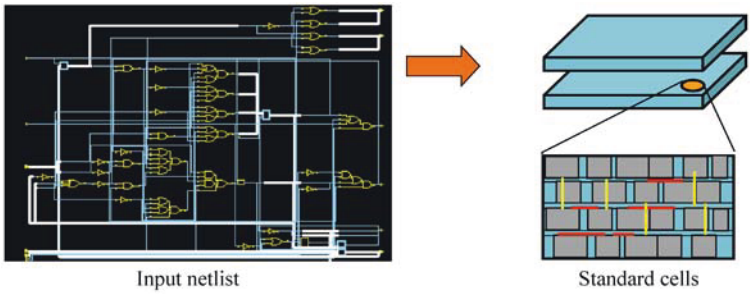


Figure 6.1 2.5-D placement problem

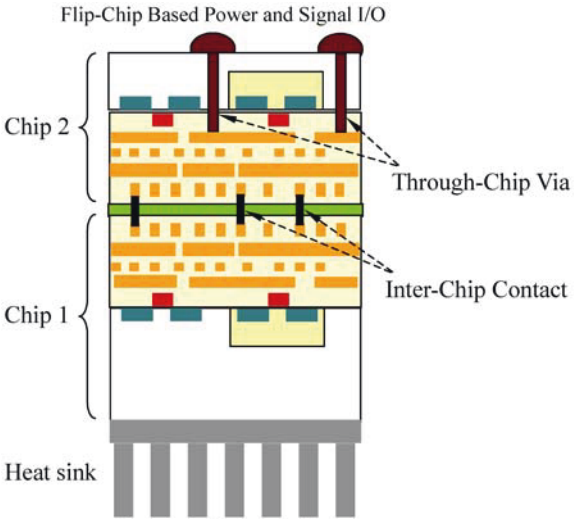


Figure 8.1 Area power I/O for 2.5-D integration

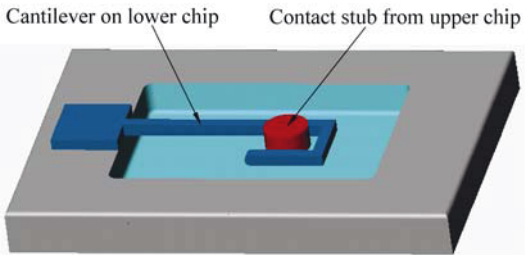


Figure 8.2 MEMS based inter-chip contact