



信息科学技术学术著作丛书

Modern ASIC Design

现代应用集成电路设计

Zhou Dian



科学出版社

Responsible Editor: Sun Fang

Copyright © 2011 by Science Press
Published by Science Press
16 Donghuangchenggen North Street
Beijing 100717, China

Printed in Beijing

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the copyright owner.

ISBN 978-7-03-031766-7 (Beijing)

国家科学技术学术著作出版基金资助出版

信息科学技术学术著作丛书

Modern ASIC Design

现代应用集成电路设计

Zhou Dian

周 电 著

科 学 出 版 社

北 京

内 容 简 介

本书基于作者在美国大学十几年教授“现代应用集成电路设计”课程的手稿整理而成, 主要内容包括应用集成电路设计流程、设计指标定义和规范、逻辑电路设计、物理设计、时间功耗性能分析及验证测试。读者需要有数字集成电路和硬件描述语言(VHDL)的基础知识。按照具体课程设置的要求, 本书可用于一个学期的教学内容, 包括应用集成电路设计流程、设计指标定义和规范、逻辑电路设计及物理设计。关于集成电路发展的前沿问题, 本书在第7章和第8章中以研究课题为背影介绍了基础知识。

本书可作为电子和计算机工作专业的大学四年级或硕士研究生教材, 也适于集成电路设计的专业人员参考阅读。

图书在版编目(CIP)数据

现代应用集成电路设计=Modern ASIC Design:英文/周电著. —北京: 科学出版社, 2011.7

(信息科学技术学术著作丛书)

ISBN 978-7-03-031766-7

I. ①现… II. ①周… III. ①集成电路-电路设计-英文 IV. ①TN402

中国版本图书馆 CIP 数据核字(2011) 第 125149 号

责任编辑: 孙 芳 / 责任校对: 张凤琴
责任印制: 赵 博 / 封面设计: 鑫联必升

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

中国科学院印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2011 年 7 月第 一 版 开本: B5 (720 × 1000)

2011 年 7 月第一次印刷 印张: 25 1/2 彩插: 2

印数: 1—2 500 字数: 496 000

定价: 120.00 元

(如有印装质量问题, 我社负责调换)

Preface

An application-specific integrated circuit (ASIC) is an integrated circuit customized for a particular application, rather than intended for general-purpose use. In the past decade, ASIC has become more mature in terms of both high design efficiency and low manufacture cost, as compared to many other available technologies. ASICs have penetrated almost every corner of our lives, from traditional computing, telecommunication, and office information processes to medical systems, automobiles, and entertainment.

ASIC design has becoming a major subject in electric and computer engineering due to its importance in modern information industries. This book is written based on the author's teaching experience and class notes on ASIC design. The intention of this book is to provide a textbook for senior level undergraduate or master level graduate students in electric and computer engineering majors. The book addresses practical issues that occur in the entire ASIC design flow, and presents the design from a system development perspective. The book can also be used as training material for engineers who would like to gain the special knowledge in this area.

Many people, especially students who attended the class in the past decade, have contributed to this book, either in shaping the content or in developing the design projects. Some of their contributions have been adopted in this book. Mr. Sagar Patel and David Hernandez-Garduno's project report is used in the Appendix. Mr. Guan-min Huang and Wei Li prepared the drafts of Chapter 5 and 8, respectively. Ms. Cathy Zhou, Belinda Zhou, Sally Liu and Mr. Dennis Liu proofread the book.

Finally, I want to thank my parents Ms. Xin-min Liu and Mr. Guang-fu Zhou for bring me up with a love for science and technology. Most importantly, I want to thank my wife Ms. Anne Y. Zhang for her persistent support of my work for more than two decades.

Zhou Dian
Dallas, Texas, USA
April 3, 2011

Contents

Preface

Chapter 1 Introduction	1
1.1 History of Integrated Circuits	2
1.2 Roadmap of IC Technology	6
1.3 ASIC	9
1.4 Design Flow	11
1.5 CAD Tools	13
1.6 An ASIC Design Project MSDAP	17
1.7 How to Use This Book	18
1.8 Summery	19
1.9 Problems	19
References	20
Chapter 2 VLSI Design Perspective and Flow	22
2.1 Introduction	23
2.2 VLSI Technology Trend	23
2.3 SoC	26
2.4 Methodology for Custom and Semi-custom IC Design	28
2.4.1 Gate array	28
2.4.2 Standard cell	30
2.4.3 FPGA	32
2.5 Design Domain and Perspective	34
2.6 Design Flow	37
2.7 Design Task	41
2.8 Summary	43
2.9 Problems	43
References	44
Chapter 3 Specification Development	46
3.1 Introduction	47
3.2 An ASIC Project MSDAP	48
3.3 An Overall View of the Specific Requirement	50

3.3.1	The required computation method by the MSDAP	50
3.3.2	Additional information for the specification	53
3.4	The System Setting	54
3.5	I/O Interface and Pins	58
3.5.1	Pins and their assignments	58
3.5.2	Signal format and waveform	59
3.6	Other Issues of the Specification	62
3.7	Summary	62
3.8	Problems	62
	References	64
Chapter 4	Architecture Design	65
4.1	Introduction	66
4.2	Datapath Structure	67
4.2.1	Single processor sequential structure	68
4.2.2	Multi-processor parallel structure	71
4.3	Functional Blocks and IPs	73
4.3.1	IP core	74
4.3.2	Functional blocks in the MSDAP architecture	75
4.4	Time Budget and Scheduling	79
4.5	A Sample Architecture of the MSDAP Project	80
4.5.1	An architecture sample	80
4.5.2	Time budget justification of the proposed architecture	96
4.6	Summary	98
4.7	Problems	98
	References	99
Chapter 5	Logic and Circuit Design	101
5.1	Introduction	102
5.2	Combinational Logics	103
5.2.1	Decoder	103
5.2.2	Encoder	106
5.2.3	Multiplexer	108
5.2.4	Arithmetic logic blocks	109

5.3	Sequential Logics	123
5.3.1	Latch and flip-flop	124
5.3.2	Shift register	136
5.3.3	Counter	138
5.3.4	FSM	144
5.4	Datapath	153
5.5	Asynchronous Circuit	156
5.6	Summery	156
5.7	Problems	156
	References	158
Chapter 6	Physical Design	159
6.1	Introduction	160
6.2	Design Rules	161
6.3	Floorplan	166
6.4	Routing	171
6.4.1	Global routing	176
6.4.2	Local routing	179
6.5	Physical Layout Verification	181
6.5.1	DRC	181
6.5.2	XOR check	182
6.5.3	Antenna check	182
6.5.4	ERC	183
6.5.5	LVS check	183
6.6	Clock Network	184
6.7	Power Network	187
6.8	Engineering Change Order	189
6.9	Package	191
6.10	Summary	193
6.11	Problems	194
	References	195
Chapter 7	Timing, Power, and Performance Analysis	198
7.1	Introduction	199

7.2	Buffer Insertion Mechanism	200
7.3	Transistor and Gate Sizing	203
7.3.1	Transistor sizing	203
7.3.2	Buffer sizing	203
7.3.3	Gate sizing	205
7.4	Timing Analysis	206
7.4.1	Static timing analysis	207
7.4.2	DTA vs. STA	210
7.4.3	Circuit simulation in STA	210
7.5	Interconnect Model and Circuit Order Reduction	211
7.5.1	Lumped RC vs. distributed RLC model	211
7.5.2	Circuit order reduction	212
7.6	Low Power Design	216
7.7	Design for Manufacture	219
7.8	High-level Synthesis	225
7.9	Performance Bound Evaluation	226
7.10	Summary	229
7.11	Problems	229
	References	230
Chapter 8	Verification and Testing	233
8.1	Introduction	234
8.2	Digital Circuits Test	236
8.2.1	Fault modeling	237
8.2.2	Fault simulation	240
8.2.3	Test generation for combinational logic	242
8.2.4	Test generation for sequential logic	245
8.2.5	ATPG using TetraMAX	250
8.3	BIST	252
8.3.1	The concept of BIST	252
8.3.2	TPG	253
8.3.3	ORA	260
8.3.4	BIST architectures	266

8.4	Scan and Boundary Scan	271
8.4.1	Digital DFT for scan	271
8.4.2	Scan chains	276
8.4.3	Digital boundary scan standard – IEEE 1149.1	281
8.5	Summary	286
8.6	Problems	286
	References	289
Appendix A	A MSDAP	290
A.1	Introduction	291
A.2	A MSDAP	291
Appendix B	A C-Program Implementing the Algorithm of the MSDAP	298
B.1	Introduction	299
B.2	The MSDAP Computation Method in C-Code	299
Appendix C	An FSM for the MSDAP Operation Mode	313
C.1	Introduction	314
C.2	An FSM for the Operation Mode and System Setting	314
Appendix D	A Sample Project MSDAP Report	326
D.1	Introduction	327
D.2	A Sample Project MSDAP Report	327

Chapter 1

Introduction

Chapter Objectives

In this chapter, you will be introduced to:

- History and roadmap of IC
- ASIC
- Design flow and tasks
- Electric design automation tools
- An ASIC design project MSDAP
- About this book

1.1 History of Integrated Circuits

In 1951, William Shockley developed the world first junction transistor, and one year later, Geoffrey W. A. Dummer published the concept of the integrated circuits (IC): “Solid block (with) layers of insulating materials”. Then in 1958 Jack Kilby (Figure 1-1) at Texas Instruments suggested the integration of circuit elements such as resistors, capacitors, and transistors into a single chip made of the same material. By September 12th of the same year, Kilby had built a simple oscillator IC with five integrated components as shown in Figure 1-2. This marked the beginning of the modern IC industry.

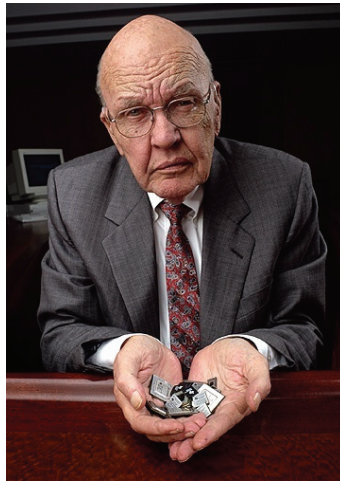


Figure 1-1 Jack St. Clair Kilby (November 8, 1923–June 20, 2005)

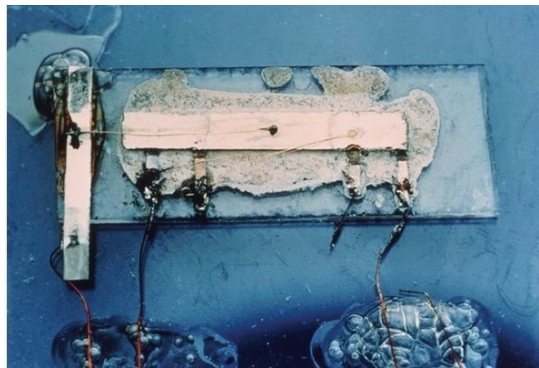


Figure 1-2 Kilby's original integrated circuits

Robert Norton Noyce (Figure 1-3) is also credited for the invention of the integrated circuits. In 1957, Noyce and several other engineers founded Fairchild Semiconductor,

where in 1959 he developed the integrated circuit. His invention was six months late, and Kilby's invention was not shared publicly at the time. In 1968, Noyce and his two Fairchild colleagues founded Intel, with Noyce as president and chief executive officer.

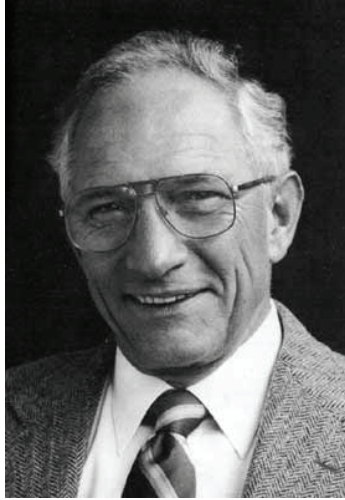


Figure 1-3 Robert Norton Noyce (December 12, 1927–June 3, 1990)

The first single chip microprocessor, based on the silicon IC, was then invented by Intel engineers Federico Faggin, Ted Hoff, and Stan Mazor in 1971. Ever since, the microprocessor has been the heart of modern electronics (Figure 1-4) (Intel 4004, 2010). Today, it can be found in almost all products, ranging from more traditional computers and cell phones, to medical equipment and automobile electronics, to name a few (ARM architecture, 2010).

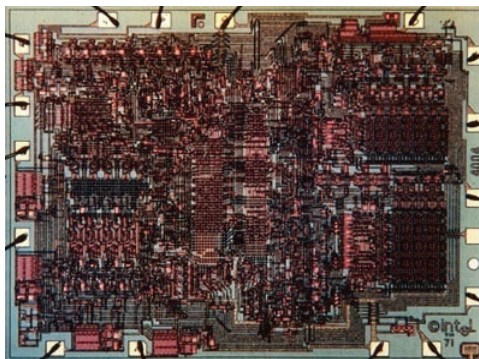


Figure 1-4 Intel 4004 microprocessor

Integrated circuits outperform vacuum tubes in most application areas in terms of size, speed and power consumption. This is attributable to the development of semicon-

ductor physics as well as twentieth century technological advancements in semiconductor device fabrication (Veendrick, 2000; Chang et al., 1996). The integration of large numbers of semiconductor transistors into a small silicon chip was an enormous improvement over the manually assembled circuits, which used discrete electronic components. The integrated circuits' mass production capability, reliability, and computer-aided design tools propelled the rapid adoption of ICs in replacing designs using discrete transistors (Rabeay et al., 2003).

Planar technology and epitaxial deposition have also helped ICs evolve. Metal-oxide-semiconductor field-effect transistors (MOSFET), transistor-transistor logic (TTL), and complementary metal-oxide-semiconductors (CMOS) were invented along the way (Sze, 1988). Today, CMOS contributes to the vast majority of all high density ICs manufactured (Weste et al., 2004; Wolf, 2002).

The advantages of ICs over discrete circuits are primarily cost and performance. Cost is low because the chips, with all of their components, are printed via a photolithography process, and millions of transistors can be manufactured and connected at the same time. Smaller feature size leads to a high performance because the components switch quicker and consume less power, due to the fact that the components are small and close together (Yeap, 1998).

The earliest version of IC, as shown in Figure 1-2, has evolved into a technological wonder. A Pentium 4 microprocessor (Figure 1-5) made in 2000 by Intel had already integrated 42 million transistors and used circuit lines of 0.18 micron feature size, a measurement of how fine we can control a line width in an IC fabrication process (Pentium 4, 2010; Rabeay et al., 2003). As of this writing, a 22nm feature size is available.



Figure 1-5 Pentium 4 microprocessor

In 2000, the importance of the IC was recognized when Kilby shared the Nobel Prize in physics with Zhores I. Alferov and Herbert Kroemer. Kilby was cited by the Nobel committee “for his part in the invention of the integrated circuit”.

The first generation of integrated circuits, called small-scale integration (SSI) circuits, contained only a few transistors. These were normally used for circuits consisting of tens of transistors. The next generation of integrated circuits, called medium-scale integration (MSI) circuits, integrated hundreds of transistors in each chip. MSI extended the integrated logic availability to counters and larger scale logic functions. Further development led to large-scale integration (LSI) in the mid 1970s, with tens of thousands of transistors per chip. LSI allowed for even larger logic functions, such as the first microprocessors Intel 4004 (containing 2000 transistors), to be consolidated into a single silicon chip (Intel 4004, 2010).

Among the most advanced integrated circuits are the microprocessors, which control everything from computers and cellular phones to digital microwave ovens (Microprocessor, 2010). In the last decade, 32-bit and 64-bit microprocessors with cache memory, floating-point arithmetic units, and multi-million transistors on a single piece of silicon have been made popular, marking the era of very large-scale integration (VLSI).

As mentioned before, integrated circuit process technology has consistently improved through smaller feature sizes over the years, allowing more circuitry to be packed into a single chip. As the feature size shrinks, almost everything improves, with cost per unit and the switching power consumption dropping, while speed increases. Small feature size improves the performance of ICs because it allows short traces, which in turn allows low power logic (such as CMOS) to be used at faster switching speed. Historically, people questioned if CMOS could compete with bipolar technology in terms of switching speed. In actuality, continuous feature size shrinking and resulting shorter channel width help maintain CMOS speed and consequently its position as the dominant technology in IC industry (Rabeay et al., 2003; Sze, 1988).

The speed and power consumption gains achieved by narrowing feature size are apparent in almost all applications. As a result, there has been fierce competition among manufacturers to use finer geometries. As a result, state-of-the-art technology for the massive production has reached feature sizes as fine as 22nm and below (ITRS, 2009).

In 1965, Gordon Moore, a director of research and development at Fairchild Semiconductor observed that “the complexity for minimum component cost has increased at a rate of roughly a factor of two per year” (Moore's law, 2010). This observation became the well-known Moore's law: The number of components per IC doubles every year. Moore's law was later amended to: The number of components per IC doubles every 18 months. Figure 1-6 shows the historic data of the real development of IC industry from 1970 to 2002 as a witness of Moor's law (ITRS, 2009). Although

there have been great concerns on the limit of ever increasing system complexity and of the accuracy of lithography, which will eventually put a stop to Moore's law, it is amazing that this rule still holds true currently.

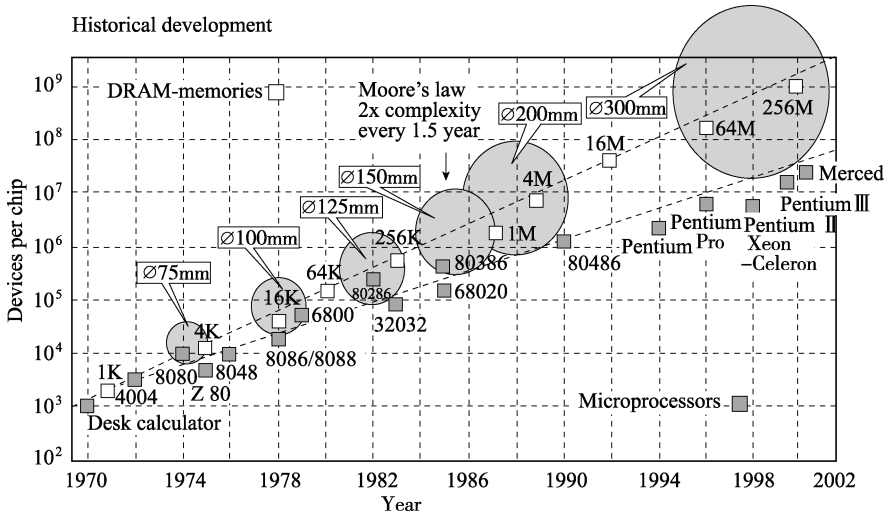


Figure 1-6 Historical development of ICs during the period from 1970 to 2002

In the figure, \varnothing is the wafer's diameter, and M is the unit for million. The density of DRAM represents the level of integration measured by process improvement, and the generation of microprocessors represents the complexity of the system.

1.2 Roadmap of IC Technology

We have seen the historic development of the IC industry. To predict and plan for the future, the International Technology Roadmap for Semiconductors, known throughout the world as the ITRS, collects and analyses the information from major IC companies,

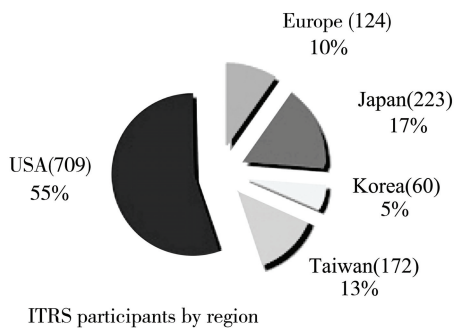


Figure 1-7 ITRS sponsors' contribution distribution

and provides a roadmap of technology milestone targets for the years to come (ITRS, 2009). The ITRS is sponsored by five leading chip manufacturing regions in the world: Europe, Japan, Korea, Taiwan, and USA as shown in Figure 1-7. Specifically, the sponsoring organizations are the European Semiconductor Industry Association (ESIA), the Japan Electronics and Information Technology Industries Association (JEITA), the Korean Semiconductor Industry Association (KSIA), the Taiwan Semiconductor Industry Association

(TSIA), and the United States Semiconductor Industry Association (SIA).

The objective of the ITRS is to ensure cost-effective advancements in the performance of the integrated circuits and the products that employ such devices, thereby continuing the health and success of this industry (ITRS, 2009). Through the cooperative efforts of the global chip manufacturers and equipment suppliers, research communities, and consortia, the roadmap teams identify critical challenges, encourage innovative solutions, and welcome participation from the semiconductor community. These teams are joining with other strategic roadmap efforts, such as electronics and nanotechnologies, so that the roadmap effort spans the spectrum of needs from basic research capabilities to product potentials.

The ITRS roadmap has been a major guide for the IC R&D objectives and an information resource (Figure 1-8). For example, the following roadmap from ITRS provides data on half-pitch and gate length (a measurement similar to feature size) for the past twenty years, and prediction of the next fifteen years.

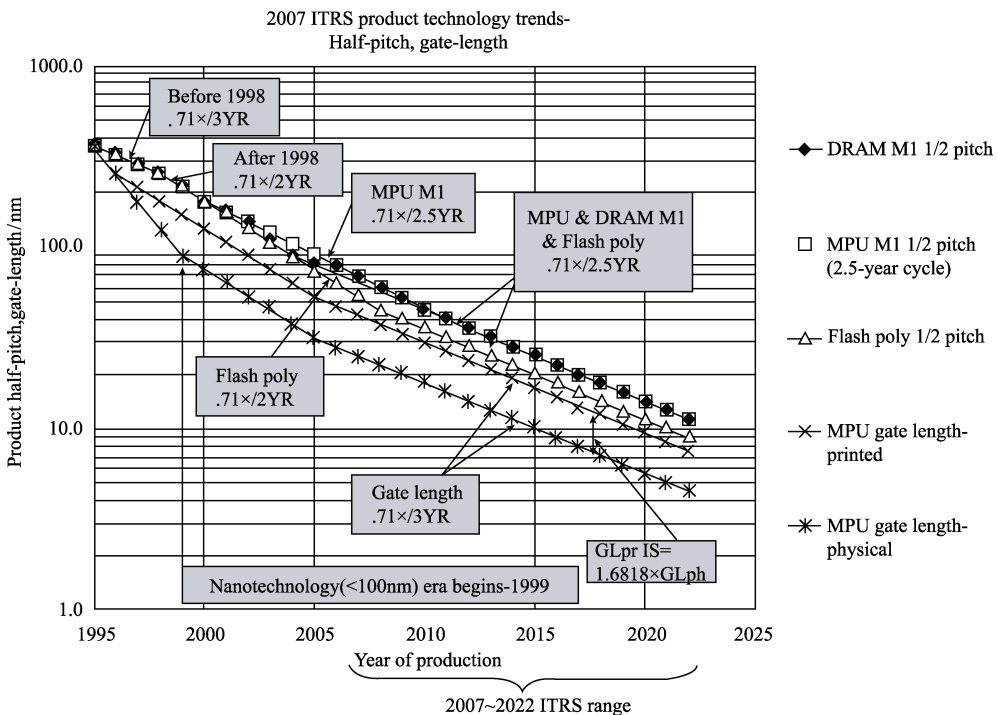


Figure 1-8 Roadmap of product half-pitch and gate length (ITRS, 2009)

To see how accurate the ITRS roadmap is, the following table compares the real historical data and that predicted by the road map. The Table 1-1 shows the historical data of the half-pitch against the ITRS prediction for DRAM products. One can see that the roadmap has predicted this technological trend fairly accurately.

Table 1-1 Roadmap versus actual trend numbers (DRAM product trend example)

Year of production	1995	1998	2000	2002	2003	2004	2006	2007	2009
Calculated trend numbers/nm	360	255	180	127.3	101	90	71.4	63.3	50.5
ITRS rounded numbers/nm	350	250	180	130	100	90	70	65	50
Year of production	2010	2012	2013	2015	2016	2018	2019	2022	
Calculated trend numbers/nm	45	35.7	31.8	25.3	22.5	17.9	15.9	11.3	
ITRS rounded numbers/nm	45	36	32	25	22	18	16	11	

ITRS also provides much more detailed information for specific classes of IC products. As mentioned above such information is very important not only for an IC company, but also for setting up research objectives. For instance, the ITRS roadmap for DRAM is shown in the following Table 1-2.

Table 1-2 DRAM introduction product generations and chip size model

Year of production	2007	2008	2009	2010	2011
DRAM 1/2 pitch (contacted)/nm	65	57	50	45	40
MPU/ASIC metal 1(M1) 1/2 pitch (f)/nm	68	59	52	45	40
MPU physical gate length/nm	25	23	20	18	16
Cell area factor/a	6	6	6	6	6
Cell area [$C_a=af^2$]/ μm^2	0.024	0.019	0.015	0.012	0.0096
Cell array area at introduction (% of chip size) §	73.52%	73.76%	73.97%	74.16%	74.30%
Generation at introduction §	16G	16G	16G	32G	32G
Functions per chip/Gbits	17.18	17.18	34.36	34.36	34.36
Chip size at introduction/ mm^2 §	568	449	711	563	446
Gbits/ cm^2 at introduction §	3.03	3.82	4.83	6.10	7.70
Year of production	2012	2013	2014	2015	
DRAM 1/2 pitch (contacted)/nm	36	32	28	25	
MPU/ASIC metal 1(M1) 1/2 pitch (f)/nm	36	32	28	25	
MPU physical gate length/nm	14	13	11	10	
Cell area factor/a	6	6	6	6	
Cell area [$C_a=af^2$]/ μm^2	0.0077	0.0061	0.0048	0.0038	
Cell array area at introduction (% of chip size) §	74.47%	74.61%	74.70%	74.83%	
Generation at introduction §	32G	64G	64G	64G	
Functions per chip/Gbits	68.72	68.72	68.72	68.72	
Chip size at introduction/ mm^2 §	706	560	444	351	
Gbits/ cm^2 at introduction §	9.73	12.28	15.49	19.55	

Moore's law mainly addresses the level of integration with respect to the number of transistors integrated into a chip and the complexity of the integrated systems. As IC application is extended into every corner of our life, we have begun to look beyond Moore's law. The following Figure 1-9 tries to explore the future from a more com-

plex perspective. At this point, there is no consensus on the limit of feature size, the technology beyond CMOS, and the broadness of application diversity (ITRS, 2009).

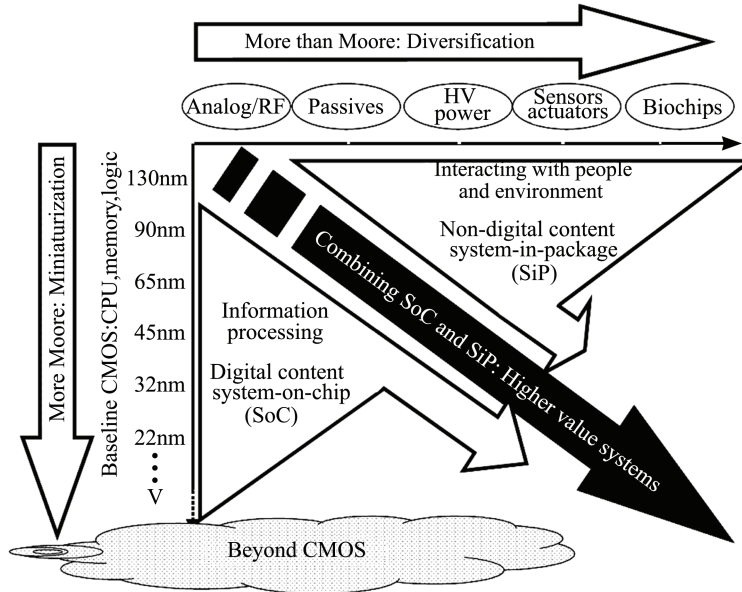


Figure 1-9 Moore's law and more

The current trend of the-state-of-the-art microprocessor technology is to use multi-core architecture, instead of increasing clock rate drastically in a single core chip. The reason is that to make an IC chip to operate at very high frequency is technology difficult and costly.

1.3 ASIC

An application-specific integrated circuit (ASIC) is an integrated circuit customized for a particular application, rather than intended for general purpose use (ASIC-SOC-VLSI design, 2011; Nekoogar, 2003). For example, a chip designed solely to run a cell phone is an ASIC while a Pentium type microprocessor or a ROM is not. In the past decade, IC technology has becoming mature in terms of both high IC design efficiency and low manufacture cost, as compared to many other available technology choices. ICs have penetrated almost every corner of our lives, from traditional computing, telecommunication, and office information processes to medical systems, automobiles, and entertainment. From this point of view, most of today's IC chips are ASICs. To peek into IC wide range applications, Figure 1-10 illustrates a few IC applications for wireless communication greater than 20GHz.

According to the type of application, development and manufacturing process, and

behavioral criteria, ASICs can be classified as:

- Full-custom ASICs: Some or all of the logic cells, circuits and layouts are specifically designed for an intended application.
- Semi-custom ASICs: All of the logic cells are pre-designed and required interconnects for specific functions are done by customizing a few mask layers.
- Non-custom ASICs: All logic cells and interconnecting wire segments are pre-designed. The chip is even packaged before any specific application. Only the switches connecting the wire segments need to be turned on (or off) according to the application logic.

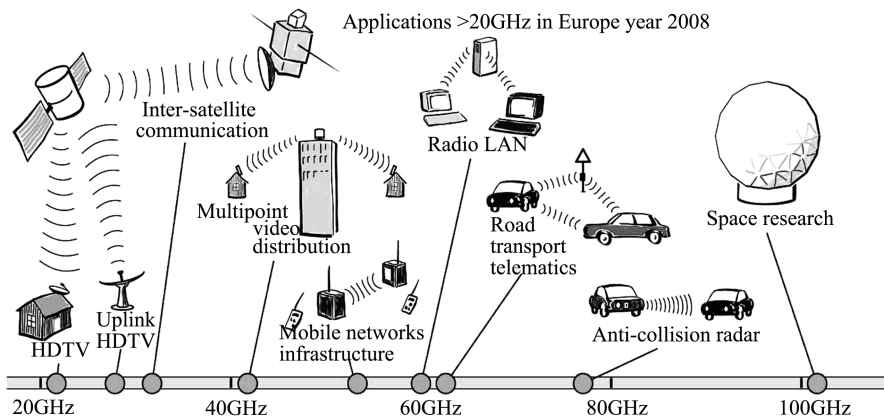


Figure 1-10 Application of ICs in wireless applications

Microprocessors are usually fully customized. However, it is traditionally not classified as an ASIC due to its special importance and many applications.

Gate arrays (GA) are a typical semi-custom ASIC where basic logic cells are placed in an array prior to the design process and one only needs to properly connect these existing cells to achieve different intended functions (Gate array, 2010). Because masks for the transistors are fixed for all gate array chips regardless of the application, the chip design and manufacture cost is lower than that of fully customized ASICs.

Field programmable gate arrays (FPGA) are a typical representative of non-custom ASICs where containing both programmable logic components and programmable interconnects (FPGA, 2010). The programmable logic components can be programmed to perform more complex combinational and sequential functions. They also include memory elements, interface logics, etc. Furthermore, programmable interconnects allows the logic blocks of an FPGA to be interconnected as needed by the system designer, like an on-chip programmable breadboard.

Fully customized ASICs are at an advantage with their higher performance and relatively smaller chip area, as compared to the semi-custom and non-custom ASICs.

Semi-custom ASICs have transistors and gates designed prior to an application and therefore only need mask design for interconnects is necessary. This results in a reduced design cycle and cost at the sacrifice of performance and silicon area, because in general there are always some gates in the chip which will not be used. Programmable logic ASICs such as FPGAs have the shortest design cycle and lowest design cost for an application. Again such an advantage is achieved at the sacrifice of performance and usage of chip area. It needs to be pointed out that performance of today's FPGAs is actually sufficient enough to meet most application requirements. State-of-the-art FPGAs integrate multi-million transistors and operates at several hundred MHz or even GHz clock rates. The only drawback is that they are usually expensive because they are not specifically tailored for a given application at hand. FPGAs are very suitable for prototype development where only a small volume is needed. In such a situation, one only needs to burn the design (netlist) into an off shelf FPGA, which avoids the heavy cost related to a customized mask set.

An ASIC design starts from a specific application, and is followed by developing the specifications, choosing the proper technology (full-custom, semi-custom or programmable logics), drawing the architecture, optimizing the logics and circuits, making physical layouts of functional blocks and interconnects, verifying the design, and testing the fabricated chips. In the following section, we shall outline the contents of each task and design flow respectively.

1.4 Design Flow

Of all aspects of an ASIC project, design flow places the heaviest demand on a design engineer and requires the closest teamwork between the customer and provider. Design flow provides the sequence of tasks and the interplay among them. Figure 1-11 shows a simplified design flow that demonstrates the basic design tasks and the process. The real design flow can be more complicated and involve more details, which will be addressed in subsequent chapters of this book. At this point, we shall outline what we do in each task and the primary relationship among them.

These basic tasks are usually interspersed in today's complex digital system design process. The modern ASIC design flow has evolved and increased in complexity, just as the circuits that are being designed have dramatically increased in complexity (Vai, 2000). The design flow is now heavily dependent on EDA tools and many of the tasks that were once carried out manually are now automated with little or no manual inter-

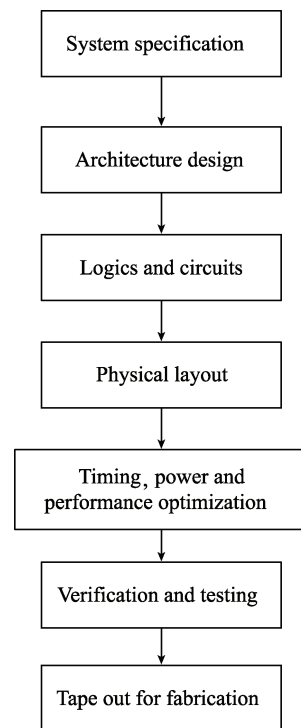


Figure 1-11 Design tasks and a simple design flow

vention. As an example, Figure 1-12 gives a comparison of a typical ASIC and FPGA design flow (Brown et al., 1996).

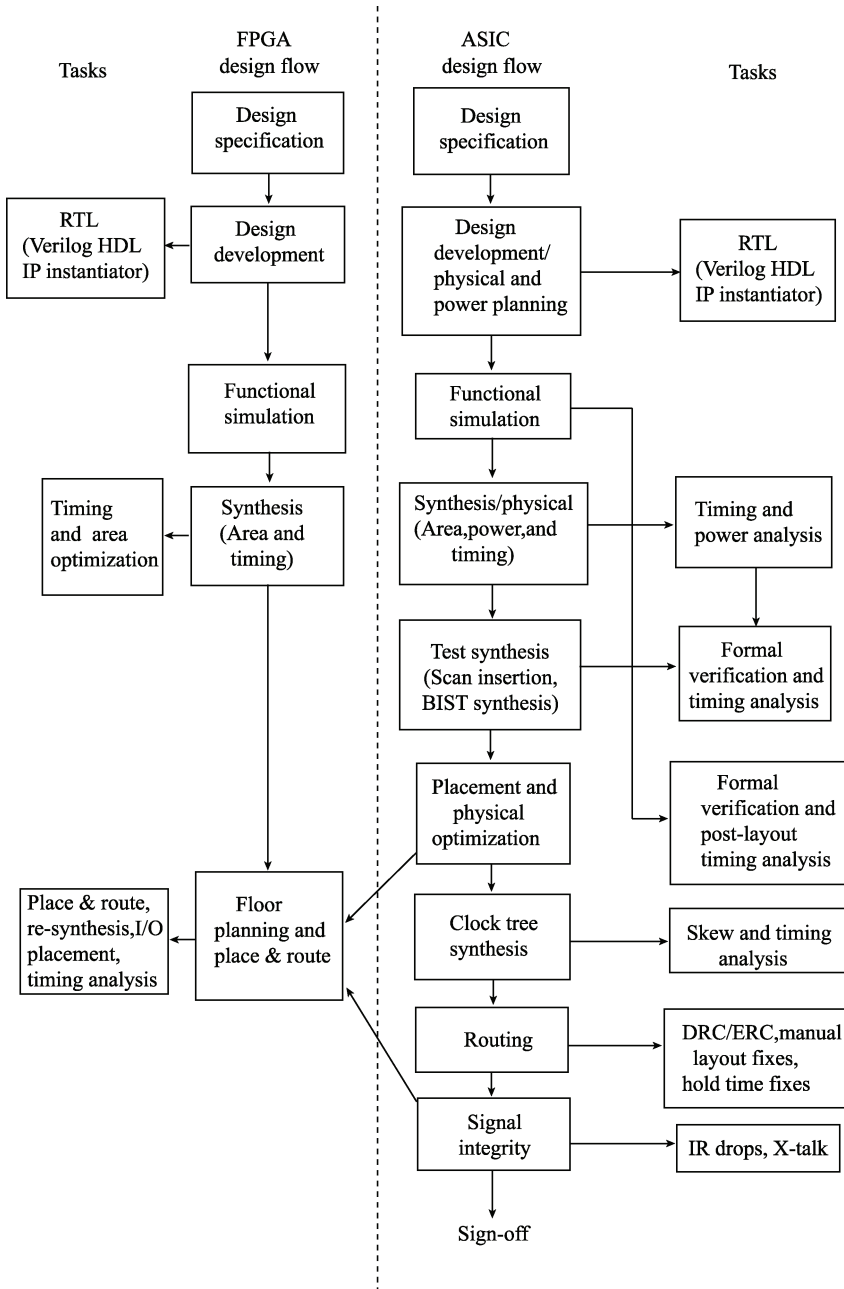


Figure 1-12 A comparison of general ASIC & FPGA design flows

The backend design of an ASIC device involves a wide variety of complex tasks, in-

cluding placement and physical optimization, clock tree synthesis, signal integrity, and routing using different EDA software tools (Routing, 2011; Sherwani, 1999). As a comparison, the physical (or backend) design of FPGA devices is very simple and is accomplished with a single software tool. This not only reduces the complexity of the design process, but also significantly reduces cost.

Design flow may vary with the chosen technology, companies' design environment, CAD tool settings and sometimes personal taste. These variations are modified to fit specific needs.

1.5 CAD Tools

CAD (Computer-aided-design) uses a wide range of computer-based tools to assist engineers, architects and other design professionals in their design activities (Rubi, 1994). CAD is used throughout the engineering process from conceptual design to layout, i.e., through designing methods to the detailed engineering and analysis of components. In the following, we mention some CAD platforms from major CAD vendors based on their product information. Readers are encouraged find more details and updated information directly from the companies' websites.

1. Cadence Virtuoso custom design platform

The Virtuoso platform from Cadence is a comprehensive system that enables design teams to deliver silicon that meets the specifications (Cadence, 2010). It includes a specification-driven environment, multi-mode simulation, accelerated layout, advanced silicon analysis, and a full-chip integration environment. The Virtuoso platform enables a meet-in-the-middle methodology that combines the speed of top-down design with the silicon accuracy of bottom-up design (Figure 1-13).

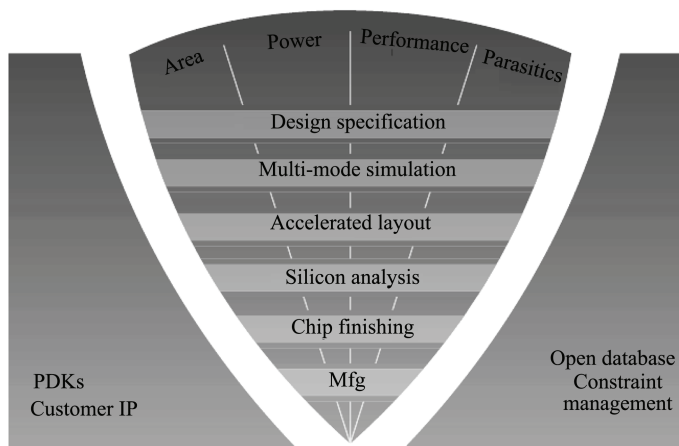


Figure 1-13 Cadence Virtuoso platform and its functional components

While platforms like Cadence Virtuoso address the entire design flow and methodol-

ogy, some special CAD packages have been developed to handle particular design tasks. For instance, the Cadence Encounter platform is a CAD package used mainly for performance-driven physical design, which takes the synthesized RTL (register transfer level) netlist from a high level synthesis tool and implements the physical layout (RTL, 2011; Baker, 2008; Coussy, 2008; Gajski et al., 1992). During this process, many physical constraints, from circuit timing and design rules to power consumption are considered. Cadence is especially well known for its physical design tools, although it offers a complete suite of IC design EDA tools. The functionality of the Encounter platform is given by Cadence as follows.

2. Cadence Encounter digital IC design platform

The Encounter platform is an integrated RTL-to-GDSII design environment (Figure 1-14). It includes a full spectrum of technologies for nanometer-scale digital design. These advanced capabilities include synthesis, testing, formal verification, physical prototyping, signal integrity and timing, routing, constraint management, yielding, power, and more. It provides a complete flow from RTL synthesis and test design through physical prototyping and partitioning to final timing and manufacturing closure.

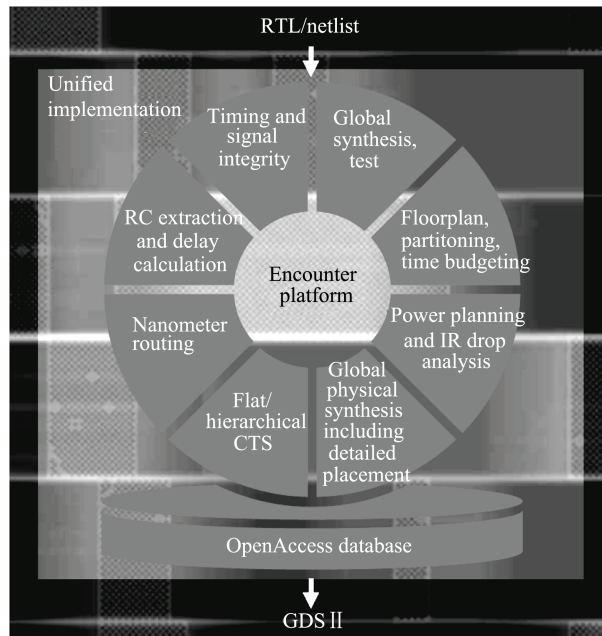


Figure 1-14 The Encounter platform

Another major EDA company is Synopsys. This company's strength is high-level synthesis, which covers design flow from specification to RTL (Micheli, 1994). Its products offer a good interface and design entry at a high level. The following is the

introduction provided by Synopsis on its product design compiler.

3. Synopsis design compiler

Synopsys has focused on delivering state-of-the-art technologies such as gate-level synthesis, RTL synthesis, pre-verified DesignWare IP, integrated power & test optimization, advanced datapath optimizations, and physical synthesis (Synopsys, 2010). Its latest innovation is design compiler ultra topographical technology that enables designers to accurately predict post-layout timing and area during RTL synthesis without the need for wire load model-based timing approximations (Figure 1-15).

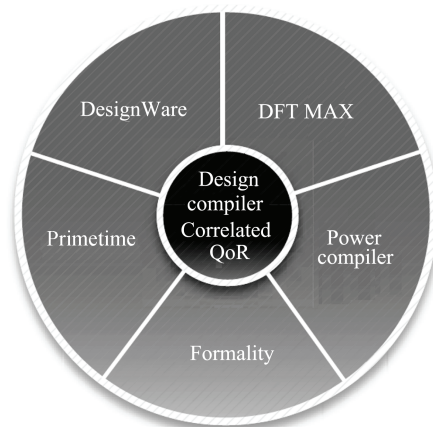


Figure 1-15 The comprehensive synthesis solution

Created for RTL designers, design compiler ultra topographical technology requires no physical design expertise or changes to the synthesis use model. By accurately predicting post-layout timing and area, this solution eliminates costly iterations between synthesis and layout, allowing results to be obtained more quickly.

At high levels of logic design, the entry point can be a behavior model described by HDL (hardware description language) or by System-C. This is quite common when designing a SoC (system-on-a-chip) using existing IPs (intellectual property) (IPs, 2010). Therefore, the ability to take the design input at system level and gluing IPs and functional blocks together is an important task. Synopsis offers a platform called “innovator” for this purpose (Figure 1-16). The functionality of this tool given by Synopsis is as follows.

4. Synopsis innovator

Synopsys innovator is a System-C based integrated development environment (IDE) for virtual platform developers to efficiently integrate, analyze and verify transaction-level models. In addition to reuse of existing IEEE 1666 System-C models with arbitrary interfaces, innovator enables the authoring of Open System-C Initiative's

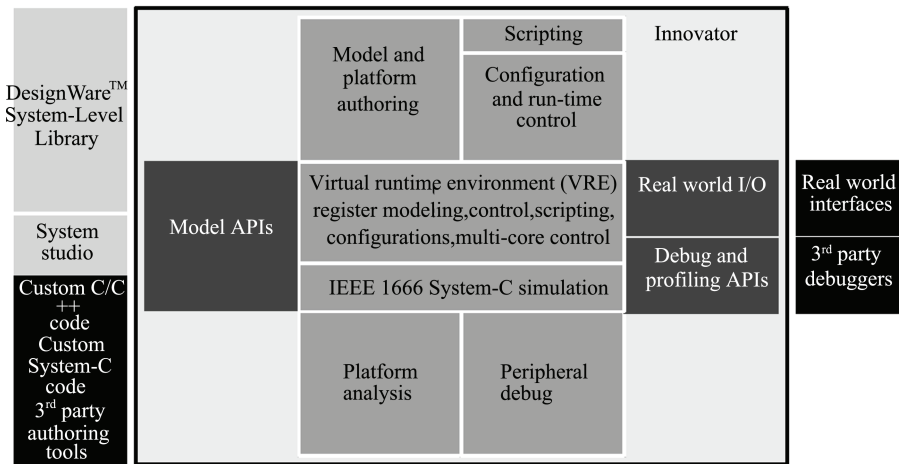


Figure 1-16 Synopsis innovator functional blocks

(OSCI) and System-C TLM-2.0 compatible models. Once a virtual platform is assembled and verified with innovator, it can be delivered to programmers as a standalone, run-time executable, seamlessly fitting into software developers' work flows. Innovator provides open interfaces for software debuggers and software development tool chains. In combination with the Synopsys DesignWare® System-Level Library of standards-based transaction-level models (TLMs), Synopsys innovator significantly increases productivity in creating, assembling and deploying virtual platforms. Innovator supports System-C loosely timed (LT) modeling for pre-silicon software development and System-C approximately timed (AT) modeling for architecture analysis as well as links to RTL verification.

5. Altera Quartus II software

Both Cadence and Synopsys tools are for custom ASIC design. There are special CAD tools for semi-custom design as well, such as Quartus from Altera for FPGA design (Altera, 2010). It is clear that such CAD tools work at high level and specifically for programmable devices. If used, we need only to obtain the RTL netlist based on the constraints posted by the pre-designed gates and connects. Figure 1-17 shows this tool's flow, which explains the features of the software and how it assists a designer in FPGA and CPLD design. The placement and routing in this case is to map the logic netlist into pre-fabricated gate arrays (also called technology mapping), and to connect them using pre-placed interconnect segments. With this CAD tool, ModelSim can provide a comprehensive simulation and debug environment for complex ASIC and FPGA designs. Support allows for multiple languages including Verilog, System_Verilog, VHDL, and System-C.

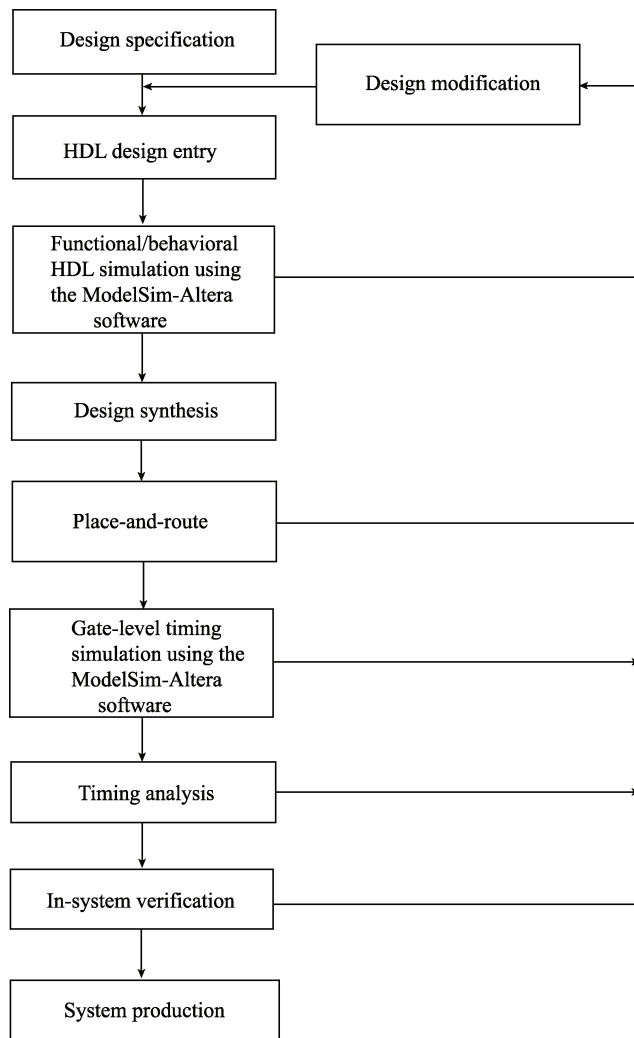


Figure 1-17 Altera design flow with Quartus II software

1.6 An ASIC Design Project MSDAP

The best way to learn ASIC design is to design a “real” ASIC chip. By going through each task in the entire design flow one not only obtains the knowledge of individual tasks, but also understands the interplay among them. Most existing textbooks mainly focus on the logic functions, circuit components, system control, interface and synthesis method, and treat them as separated subjects. However, the subjects of properly defining system settings and connecting individual design tasks have not received adequate attention.

This book presents a “real” ASIC design project (A mini stereo digital signal processor: MSDSP) and guides the reader through the whole design process (Appendices A-D). The project is to design an ASIC chip which implements an IRF filter used in a hearing-aid device. The device is intended to be disposable, and therefore it needs to be low power and inexpensive at the same time.

In order to achieve the objective of low power and low cost the application requires the use of a specific implementation method. Thus designers have the opportunity to “understand” a custom specific requirement, which usually posts a challenge for many ASIC designers. In addition, this project requests the designer to develop a complete specification based on an application about which information is usually “incomplete” from a VLSI design perspective. Commonly, most applications are presented with application field terminologies, and terms such as clock rate, supply voltage, and interface protocol are usually not used. Thus this ASIC project offers a special value for readers to mimic a relatively real design environment (Hernandez-Garduno 2009).

A complete VHDL code for this ASIC project is provided in the book (Appendix D). The code is by no means optimal but serves as a reference. The interested reader can run this sample code in any standard ModelSim simulator. It can also be used with a synthesis tool and a cell library to generate a circuit level design. If a process related cell library is available, the reader can also use this code to generate a chip physical layout. With the physical layout at hand the reader can further abstract parasitic parameters such as interconnect resistance, capacitance, and inductance, and add these parasitic parameters into the netlist for circuit level simulation, which is considered accurate and close to the real fabricated circuit performance.

This ASIC project connects the discussions of each topic presented in the book. It helps to present the subject in a more concrete and specific way.

1.7 How to Use This Book

This book is written for senior level undergraduate students or master level graduate students as a textbook in the area of VLSI or ASIC design. It can also be used as a training material for engineers who would like to gain knowledge in this area. In all cases, readers should have the basic knowledge of digital circuit, entry level VLSI design, and hardware description language VHDL (Brown et al., 2008; Armstrong et al., 2000).

The book consists of total eight chapters and one appendix, as shown in Figure 1-18. To carry out the project MSDAP, the reader should have a complete suite of Cadence, Synopsis or similar design environment. An industrial process related cell library is necessary if layout and circuit level timing analysis are needed. A complete VHDL code of the implementation of MSDAP (mini stereo digital audio processor) is given in Appendix D.

The book chooses not to discuss how to write HDL code, since it is not the focus of the book and basic material on this subject is abundant. Readers can get comprehensive information on VHDL and its application from (Van der Spiegel, 2011; Chu, 2006; Armstrong, 2000; Naylor et al., 1997).

The internet has also become a major information resource. Many references and introductory materials in this book are actually obtained from the web sites. Reader can directly check those websites for more detailed information, and furthermore can easily use them as a starting point to find out more.

At the end of each chapter, there are problems designed to support and enhance the discussed subject in the book.

There are several ways to use this book based on the reader's particular needs. The simplest one follows the flow shown in Figure 1-18, which as mentioned above should be a full semester's worth of course teaching material. Chapter 5 can be skipped over if readers have adequate knowledge of digital logics and circuits, and their interest is in system level design and use of VLSI CAD tools. Chapter 7 and 8 can also be omitted if these advanced topics do not fit into a one semester teaching schedule. They can be covered in a separate short course or seminar. In this case, the book can be trimmed as a training material for special topics, such as VLSI layout, timing/performance analysis, and verification/testing. The relevant chapters can be selected accordingly.

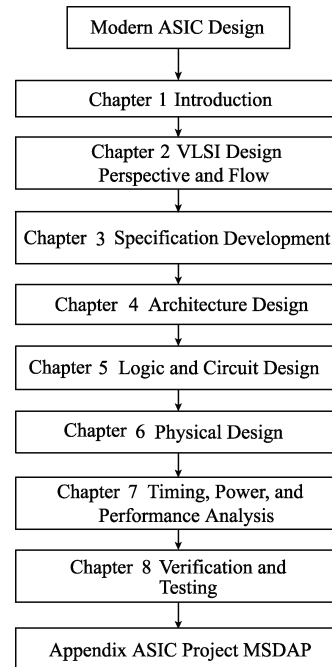


Figure 1-18 Book main contents and sequence of presentation

1.8 Summery

In this chapter, we have presented the invention of the IC and its history. As pointed out, the roadmap from ITRS will be a major information source predicting VLSI future development trend. Capability of using modern CAD tools is an essential skill for any digital circuit designer. In the following chapter, we shall explore VLSI design perspective and develop its design flow.

1.9 Problems

- (1) Write a survey of digital IC design CAD tools from the major EDA companies.
- (2) Explain the difference between the digital, analog, and mixed signal circuits. Give

an example of each type.

- (3) Suggest a few new areas, in addition to the traditional ones such as computers, communications and home electronics, in which ICs can be used in the near future. Use an example to justify your suggestion and significance.
- (4) What is a specification of an IC design project? Suggest the basic contents in a specification and give an example as complete as possible.
- (5) What is VHDL and give an example. Explain where a designer uses VHDL?
- (6) What is Verilog and give an example. Explain where a designer uses Verilog?
- (7) What is System-C, and how it is used in the digital system design?
- (8) What is an IP in digital system design?
- (9) What is SoC? What are the challenges encountered in the design of an SoC?
- (10) Study the roadmap, and name and explain a few challenges to today's digital circuit design.
- (11) Study the roadmap, and name and explain a few challenges in the nanometer scale IC manufacture process.
- (12) Explain the term “design-for-manufacture” (DFM) in today's IC design. Give an example.
- (13) Review the basics of VHDL and write a VHDL code of a D flip-flop.
- (14) Write a VHDL code of a 4-bit adder.
- (15) Study why the trend of microprocessor development is to use multi-core technology instead of to increase clock rate drastically.
- (16) What are the concerns of after Moore's law?

References

- Altera. 2010. <http://www.altera.com/>.
- ARM architecture. 2010. http://en.wikipedia.org/wiki/ARM_architecture.
- Armstrong J R. 2000. VHDL Design. 2nd Edition. Upper Saddle River: Prentice Hall.
- Armstrong J R, Gray F G. 2000. VHDL Design Representation and Synthesis. Upper Saddle River: Prentice Hall.
- ASIC-SOC-VLSI design. 2011. <http://asic-soc.blogspot.com/>.
- Baker R J. 2008. CMOS: Circuit Design, Layout, and Simulation. 2nd Edition. New York: Wiley.
- Brown S, Rose J. 1996. Architecture of FPGAs and CPLDs: A tutorial. IEEE Design and Test of Computers, 13(2): 42–57.
- Brown S, Vranesic Z. 2008. Fundamentals of Digital Logic with VHDL. New York: McGraw-Hill.
- Cadence. 2010. http://en.wikipedia.org/wiki/Cadence_Design_Systems.