

# Verilog HDL

---

## 数字系统设计

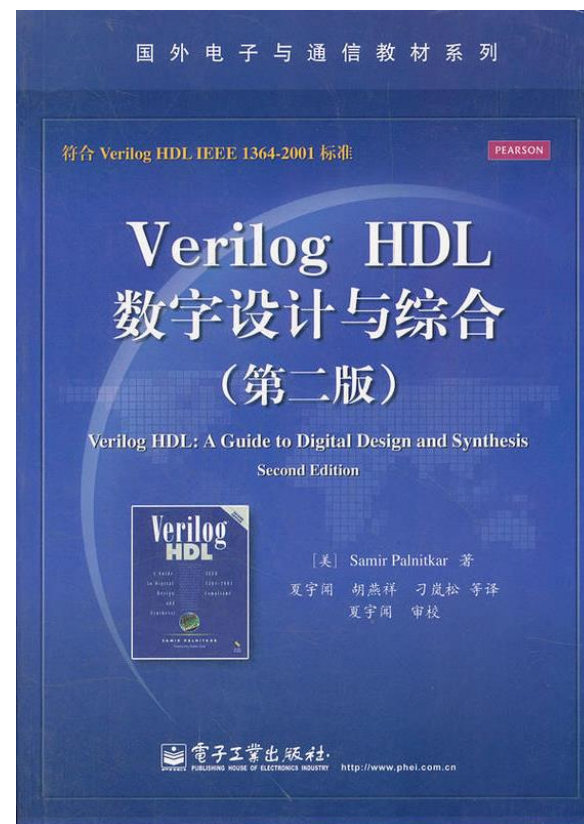
宦 飞

*E-mail: [huanfei@sjtu.edu.cn](mailto:huanfei@sjtu.edu.cn)*

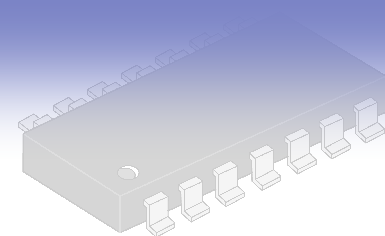


## □ 教材：

- 《Verilog HDL 数字设计与综合》，第2版
  - ◆ (Verilog HDL – A Guide to Digital Design and Synthesis, Second Edition)
  - ◆ Samir Palnitkar 著
  - ◆ 夏宇闻等译
  - ◆ 电子工业出版社

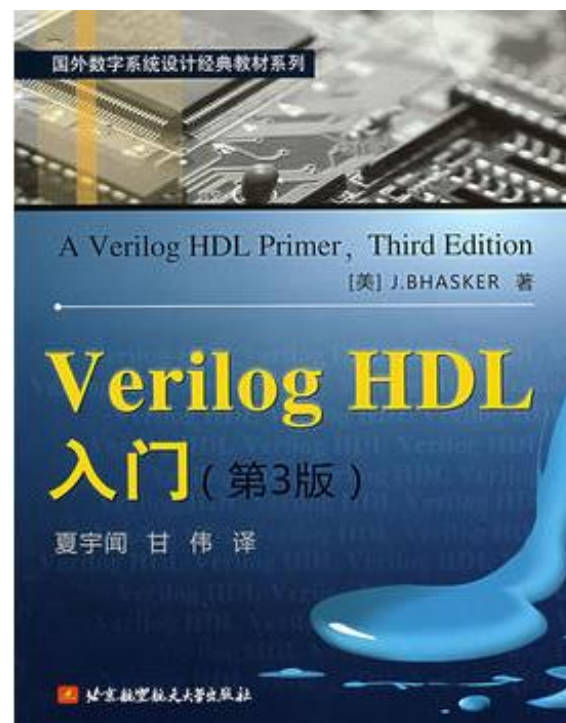
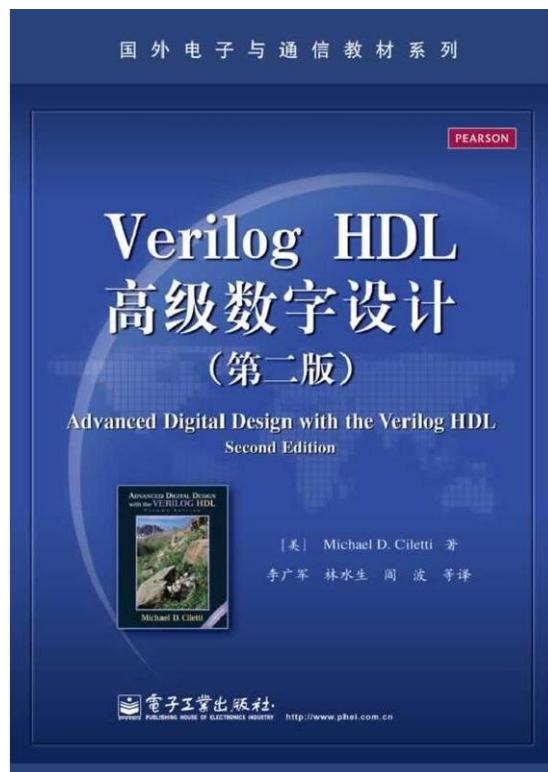


# 教学参考书



## □ 参考文献

- 《Verilog HDL 高级数字设计》（第2版），Michael D. Ciletti 著
  - ◆ 电子工业出版社
- 《Verilog HDL 入门》（第3版），J. Bhasker 著
  - ◆ 北京航空航天大学出版社..



# 用于教学的设计工具

## □ 仿真工具

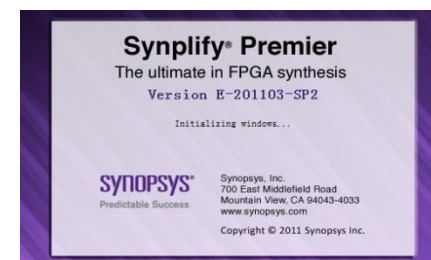
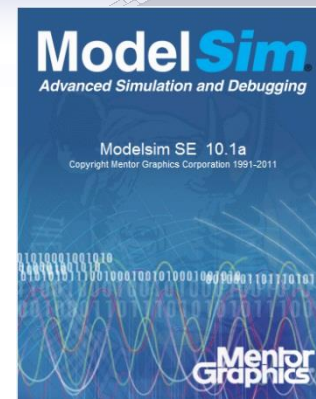
- ModelSim SE Plus 10.1a or later (v10.7c)
  - ◆ 最流行的仿真软件
- Aldec Active HDL 9.1 or later (v11.1)
  - ◆ Windows 系统上的非常流行的集成开发环境
    - ——作为另一种仿真工具使用

## □ 集成开发工具

- Xilinx ISE 10.1i ~ 14.7 —— 14.7 最后版本
- Xilinx Vivado HLx 2020.1 —— 最新版本
- Intel Quartus Prime Pro v20.2 —— 最新版本

## □ 综合工具

- Synplify Pro E-2011.03-SP2 or later
  - ◆ Windows 平台下比较流行的综合工具之一
- Precision Synthesis RTL 2012a.10 or later





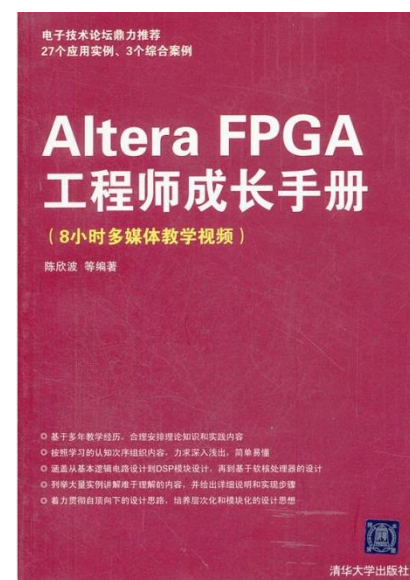
# EDA工具学习参考书

## □ 推荐参考书：

- 《Xilinx FPGA 开发实用教程》（第2版）
  - ◆ 作者：徐文波，田耕
  - ◆ 电子工业出版社
  - ◆ 适用：使用Xilinx ISE 集成开发系统（ISE 13.X）
- 《Altera FPGA 工程师成长手册》
  - ◆ 作者：陈欣波等
  - ◆ 清华大学出版社
  - ◆ 适用：使用 Altera Quatus II 集成开发系统

## □ 参考资料

- 仿真、综合工具的在线文档
- 集成开发系统（Xilinx、Intel）的在线文档
  - ◆ <http://www.xilinx.com/>
  - ◆ <https://www.intel.com/.../products/programmable/fpga.html>
- 教材后附录中推荐的各EDA网站中提供的各类文献



# 课件和工具下载

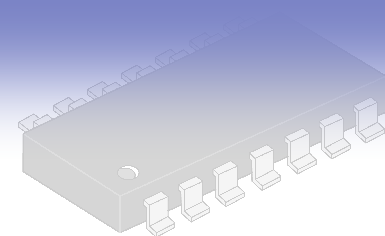


❑ ftp://public.sjtu.edu.cn



- ❑ 用户名: huanfei
- ❑ 密 码: huanfei

# 课件和工具下载



❑ <ftp://huanfei:huanfei@public.sjtu.edu.cn/DSD/>

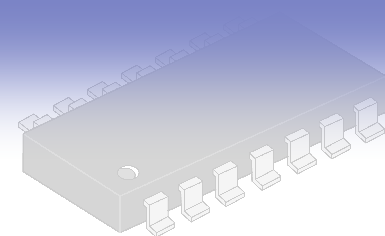
/DSD/ 的索引

应用

[\[上级目录\]](#)


名称	大小	修改日期
<a href="#">DSD_Classroom_test/</a>		2020/8/30 下午7:24:00
<a href="#">DSD_Homework/</a>		2020/8/30 下午7:24:00
<a href="#">DSD_Slides/</a>		2020/8/30 下午7:24:00
<a href="#">DSD_Textbook_&amp;_Reference/</a>		2016/9/8 上午8:00:00
<a href="#">EDA_tools_for_DSD/</a>		2016/9/8 上午8:00:00


# 课件和工具下载




❑ <ftp://huanfei:huanfei@public.sjtu.edu.cn/DSD/>

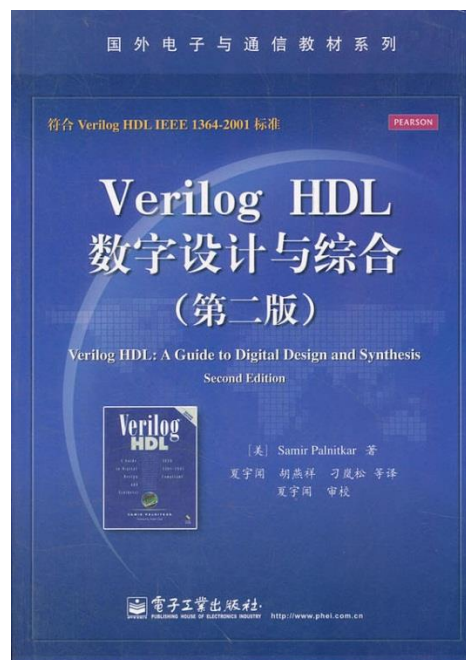
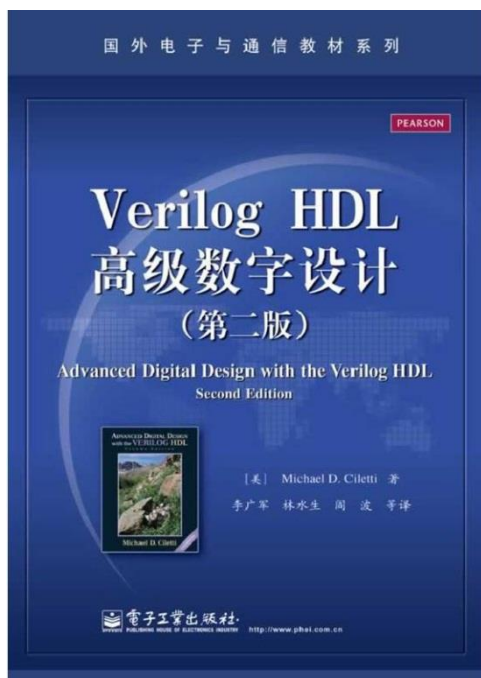
❑ 目录: DSD\_Textbook\_&\_Reference

 [上级目录]

 Book\_Advanced\_Digital\_Design\_with\_the\_Verilog\_HDL\_2nd\_edition.pdf

 Book\_Verilog\_HDL\_A\_Guide\_to\_Digital\_Design\_and\_Synthesis\_Second\_Edition.pdf

 FPGA\_Architecture\_Survey\_and\_Challenges\_NOW.pdf



Foundations and Trends® in  
Electronic Design Automation  
Vol. 2, No. 2 (2007) 125-253  
© 2008 I. Kuon, R. Tessier and J. Rose  
DOI: 10.1563/1530-090X00005

**now**  
the essence of knowledge

## FPGA Architecture: Survey and Challenges

Ian Kuon<sup>1</sup>, Russell Tessier<sup>2</sup>  
and Jonathan Rose<sup>1</sup>

<sup>1</sup> The Edward S. Rogers Sr. Department of Electrical and Computer  
Engineering, University of Toronto, Toronto, ON, Canada,  
{ikuon, jayar}@ecoc.utoronto.ca

<sup>2</sup> Department of Electrical and Computer Engineering, University of  
Massachusetts, Amherst, MA, USA, tessier@ecs.umass.edu

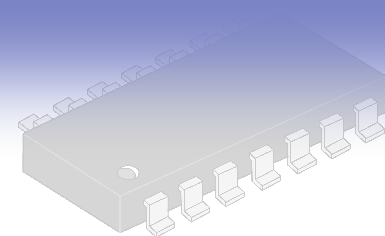
### Abstract

Field-Programmable Gate Arrays (FPGAs) have become one of the key digital circuit implementation media over the last decade. A crucial part of their creation lies in their architecture, which governs the nature of their programmable logic functionality and their programmable interconnect. FPGA architecture has a dramatic effect on the quality of the final device's speed performance, area efficiency, and power consumption. This survey reviews the historical development of programmable logic devices, the fundamental programming technologies that the programmability is built on, and then describes the basic understandings gleaned from research on architectures. We include a survey of the key elements of modern commercial FPGA architecture, and look toward future trends in the field.

FPGA技术综述(119页)



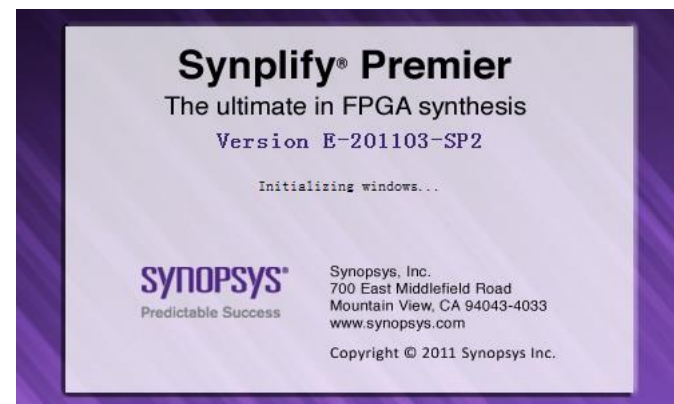
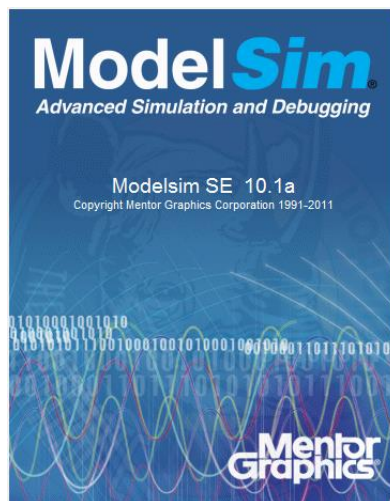
# 课件和工具下载



❑ <ftp://huanfei:huanfei@public.sjtu.edu.cn/DSD/>

❑ 目录: EDA\_tools\_for\_DSD

- Aldec\_Active\_HDL\_9\_1/
- ModelSim\_SE\_10\_1a\_X86/
- Modelsim\_SE\_10\_1c\_x64/
- ModelSim\_Tutorial\_Software\_Version\_10\_1a.PDF
- ModelSim\_User's\_Manual\_Software\_Version\_10\_1c.pdf
- Synopsys\_Synplify\_vE\_2011\_03\_sp2/

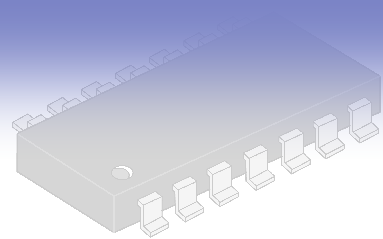


# 课程考核和成绩构成



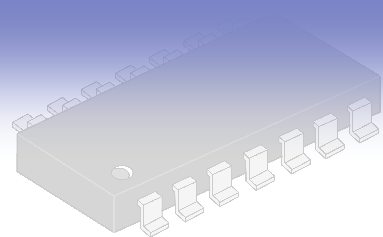
- ❑ 期末考试：70%
- ❑ 平时作业：25%~30%
- ❑ 平时记录：1%~5%

# 本课程的教学目标



- ❑ 基于 Verilog 和 PLD 设计数字系统的方法
- ❑ Verilog 是一种的硬件描述语言（Hardware description Language, HDL）
  - ❑ 用于对数字硬件系统进行设计和建模
- ❑ PLD——可编程逻辑器件（ Programmable logic device ）
  - ❑ 在EDA(Electronic Design Automation)工具支持下，使用Verilog和PLD可以快速设计、验证和实现数字系统

# 本课程的的教学内容



## □ 学习内容

- 简单数字系统（基本数字电路模块）设计和建模
- 简单数字系统仿真、验证方法
- 用于FPGA设计和开发的常用 EDA 工具的使用方法
  - ◆ 仿真工具
  - ◆ 综合工具
  - ◆ 集成开发系统和设计环境

## □ 特点

- Verilog HDL 编程实现基本电路 —— 简单
- 按照数字电路工作的方式编程实现算法 —— 非常困难
- 根据FPGA的构成和资源有效地实现算法 —— 极其困难
  - ◆ FPGA——Field-programmable gate array，现场可编程门阵列（一种可编程逻辑器件）

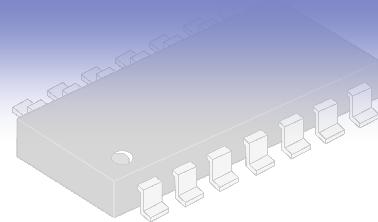


# 第 1 章

## 概 述

# 内容

---



- ❑ 数字系统设计
- ❑ 硬件描述语言
- ❑ 基于硬件描述语言的数字电路设计方法
  - ❑ 基于 FPGA 的数字电路设计

# 引言

## 通信基础设施

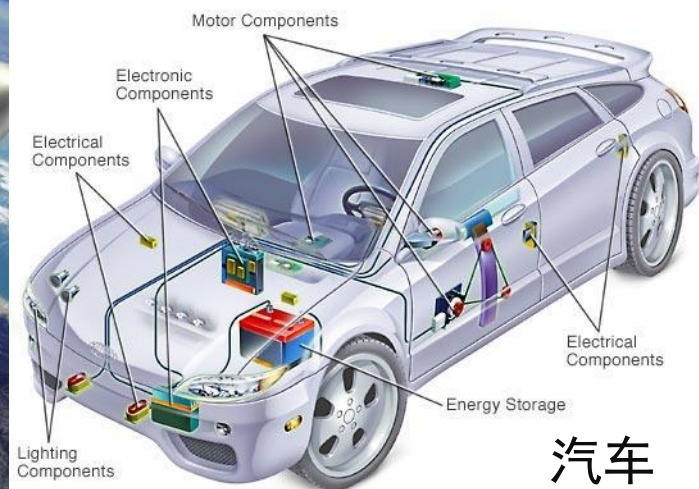
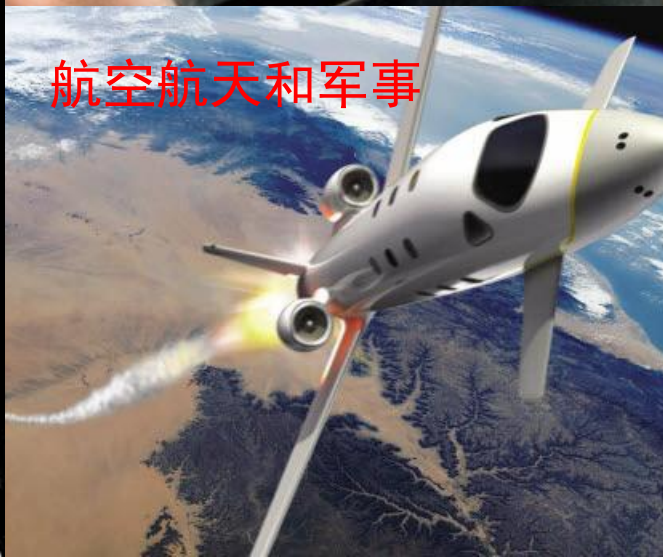
□ 几乎所有电子系统都是数字系统



消费电子产品



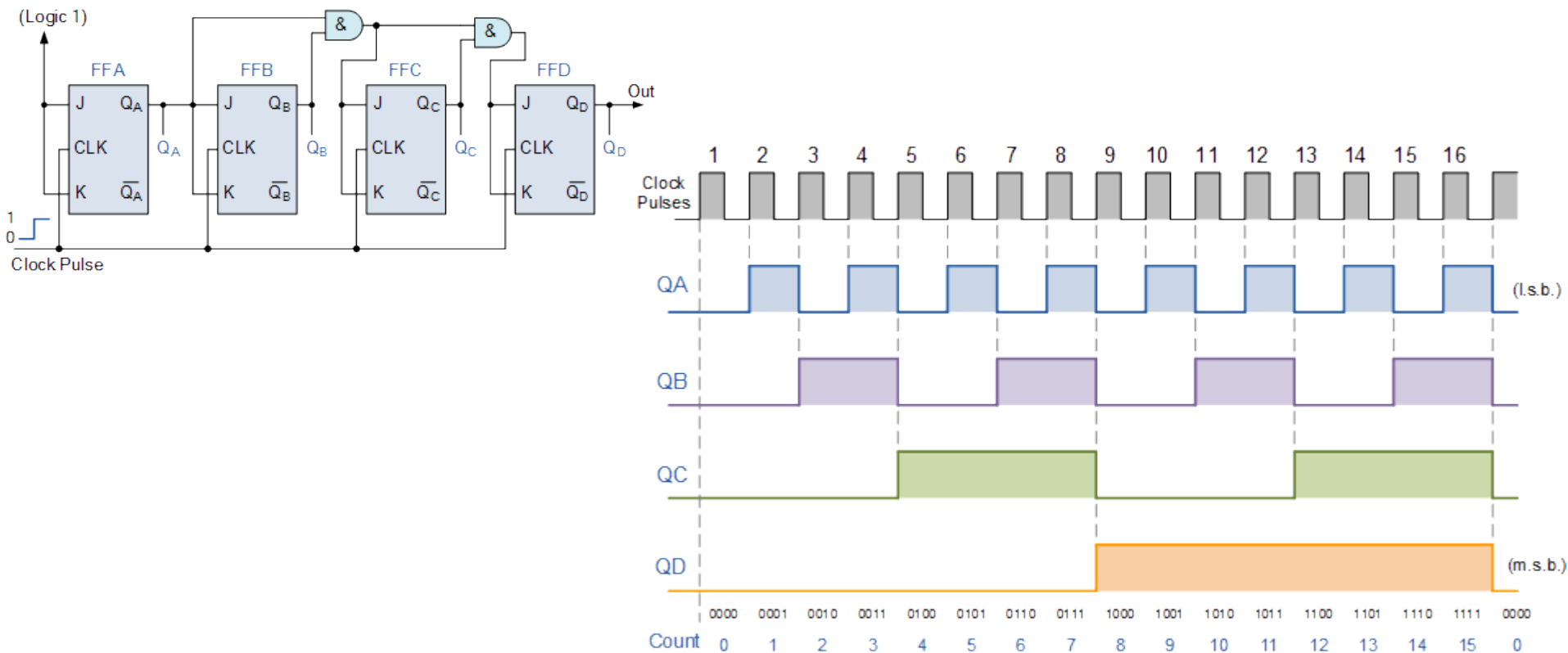
航空航天和军事



汽车

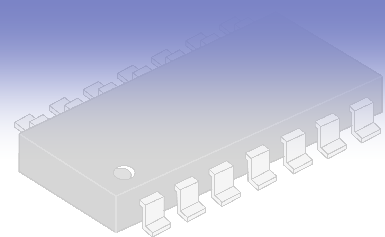
## ❑ 数字系统——同步数字硬件系统（ Synchronous Digital Hardware Systems ）

- ❑ 同步（Clocked）——系统中的一切改变都是在一个全局系统时钟的控制下，在同一时刻发生
- ❑ 数字——系统中的所有输入/输出和内部信号都是离散的数字值





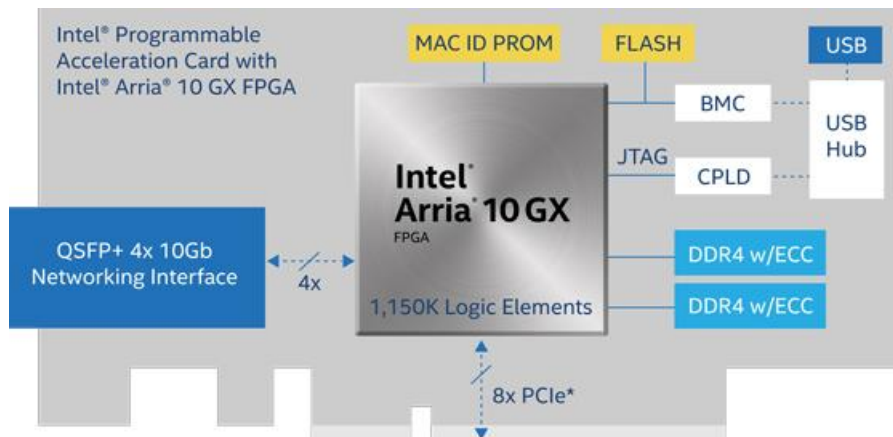
# 数字系统



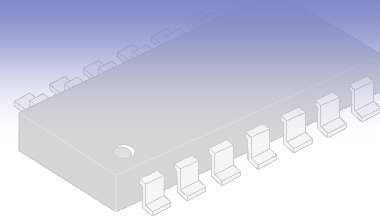
## ❑ 数字系统是信息技术的核心——无处不在

- ❑ 信息技术就是以数字系统为基础，使用**编码数据**进行计算
- ❑ 例、各种以数字集成电路为核心的电子系统
  - ◆ 台式PC、笔记本电脑、手机
  - ◆ 数字电视播放/接收设备
  - ◆ MP3、MP4播放设备
  - ◆ 测量、测试仪器 —— 数字示波器、频谱分析仪、逻辑分析仪
  - ◆ 控制装置 —— 基于DSP、MCU（单片机）的智能控制器
  - ◆ 专用实时信息、数据处理设备

## Intel Deep Learning Inference Accelerator

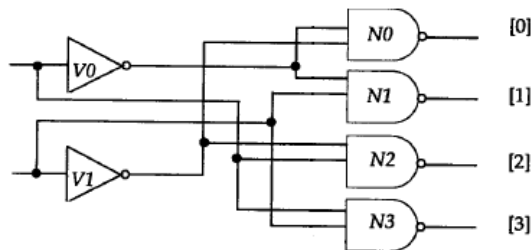


# 数字系统设计

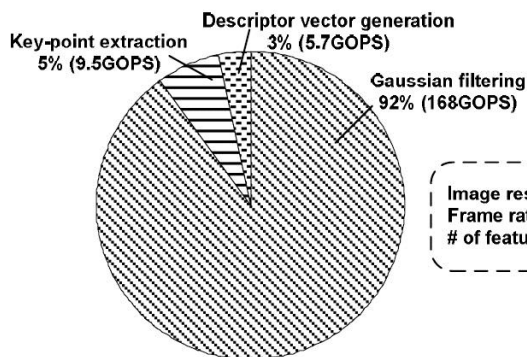


## 数字集成电路设计

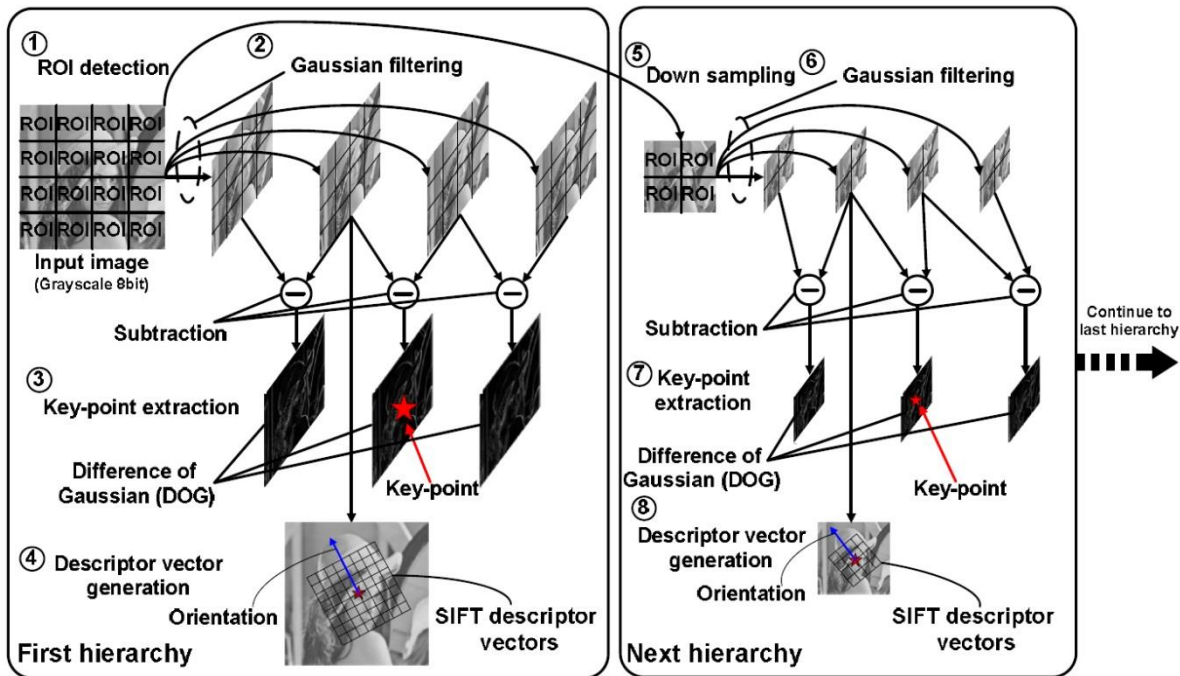
- 从非常简单的基本数字电路模块一直到极为复杂的各种处理器
- 如数字信号处理器、专用图像处理器



2-4译码器



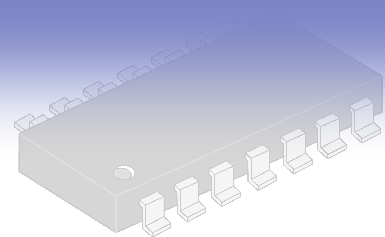
Parameter  
Image resolution : HDTV (1920 × 1080) [pixels]  
Frame rate : 30 [frame / second]  
# of feature point : 3375 [points / frame]



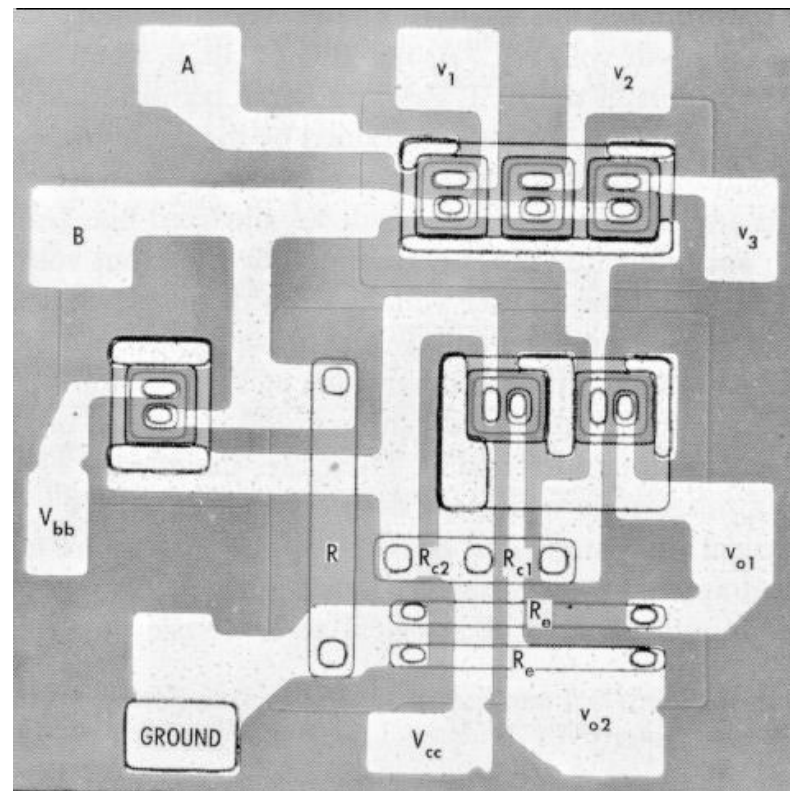
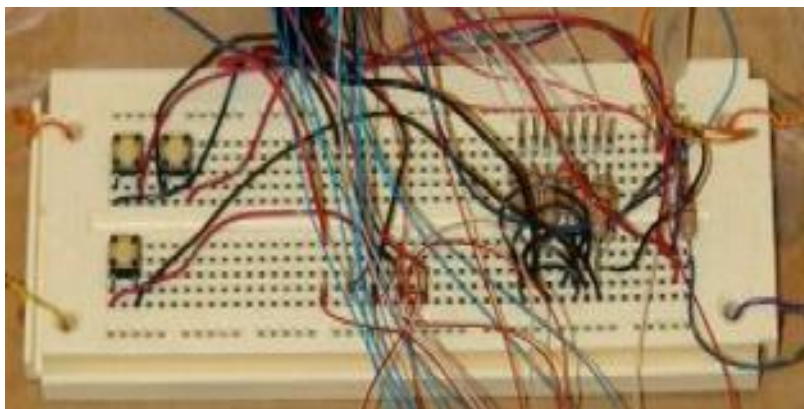
基于FPGA实现的产生 SIFT descriptor 计算流程  
SIFT —— Scale Invariant Feature Transform

软件计算SIFT descriptor负载分析,  
Gaussian 滤波占整个计算量的 90%

# 数字系统设计发展（1）



- ❑ 数字集成电路设计大致分为三个发展阶段
- ❑ 人工设计阶段
  - ❑ 早期的集成电路IC
  - ❑ SSI (**Small-Scale Integration**)
    - ◆ 包含几个到几十个晶体管
  - ❑ MSI (**Medium-Scale Integration**)
    - ◆ 包含几百个晶体管
- ❑ 特点
  - ❑ 设计和测试全部由人工完成



ECL 3-input Gate Motorola 1966

ECL: Emitter-coupled logic (射极耦合逻辑)



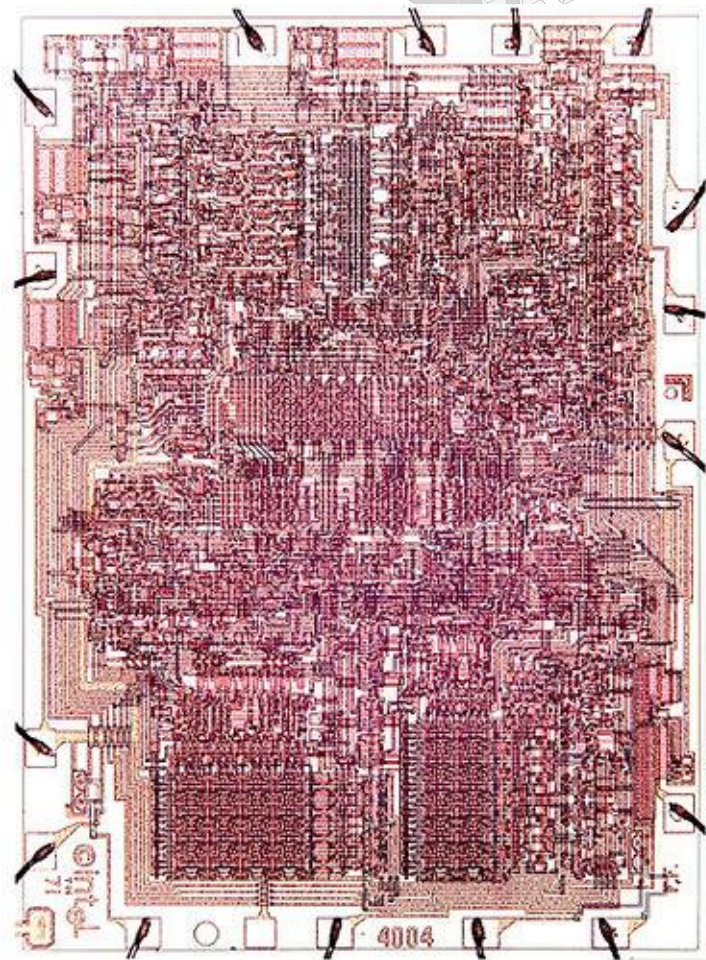
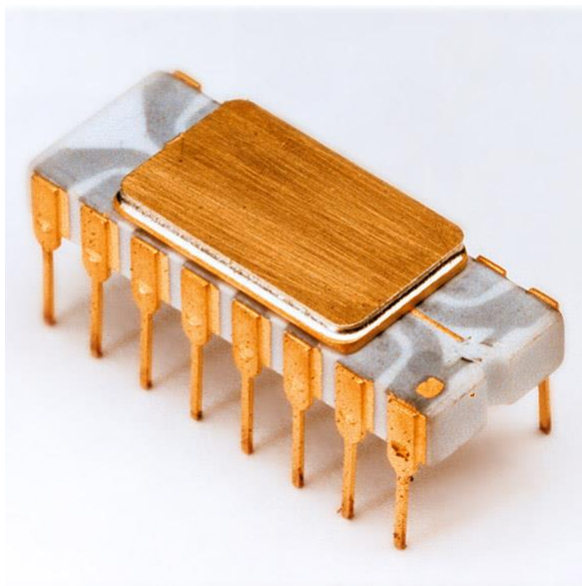
# 数字系统设计发展 (2)

## □ CAD 阶段开始

- Computer Aided Design
- LSI (**L**arge-**S**cale **I**ntegration)
  - ◆ 包含几千个晶体管

## □ 特点

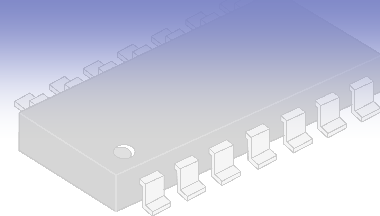
- 开始借助计算机完成某些阶段的设计
- 在计算机上完成电路原理图、版图设计
- 测试还是手工完成



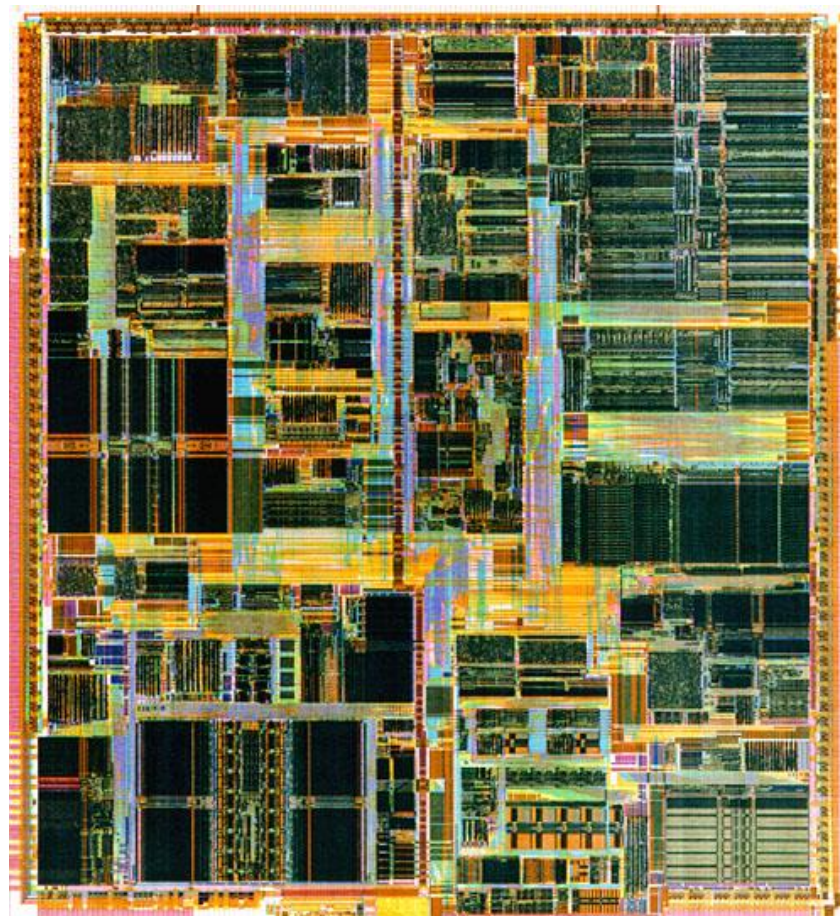
*Intel 4004 Micro-Processor* , 1971  
1000 transistors, 1 MHz operation



# 数字系统设计发展 (3)

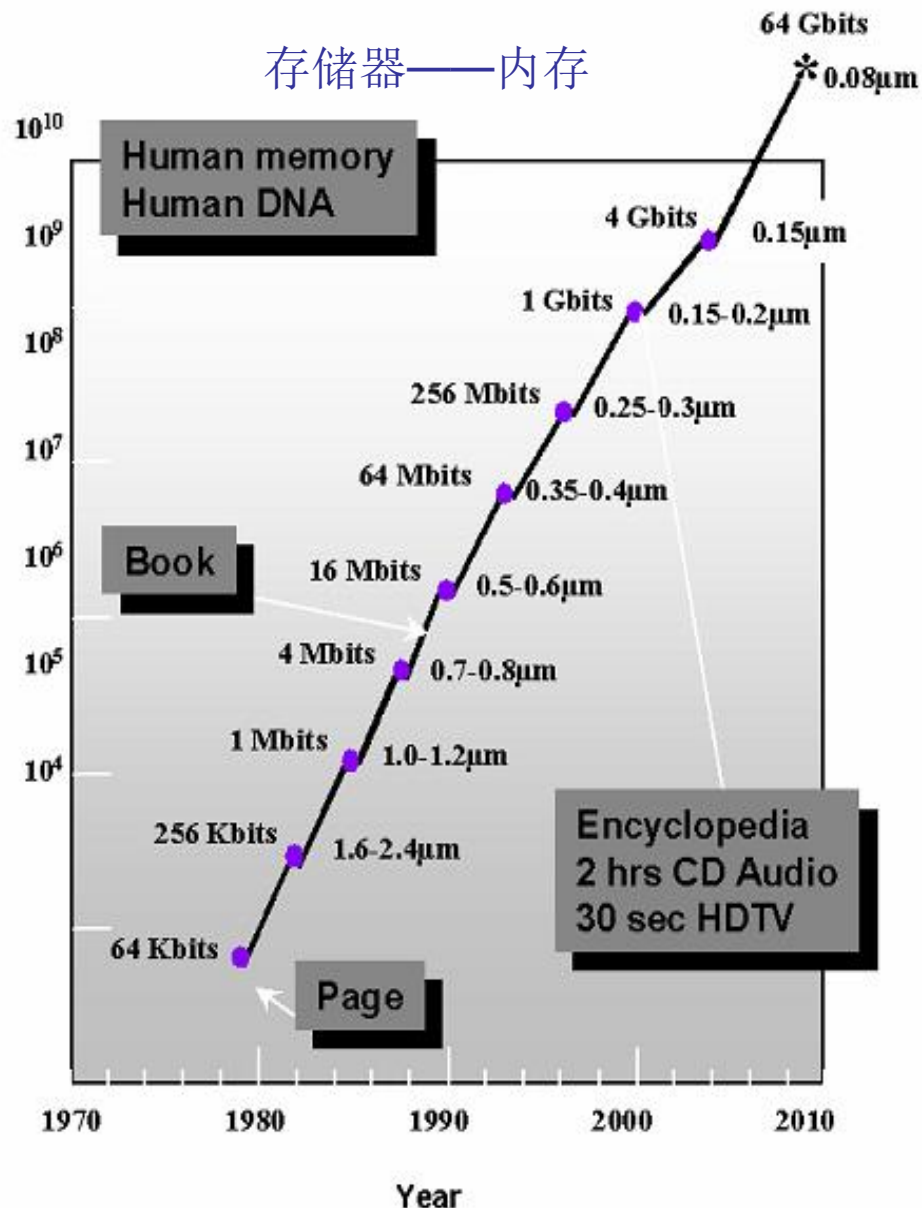
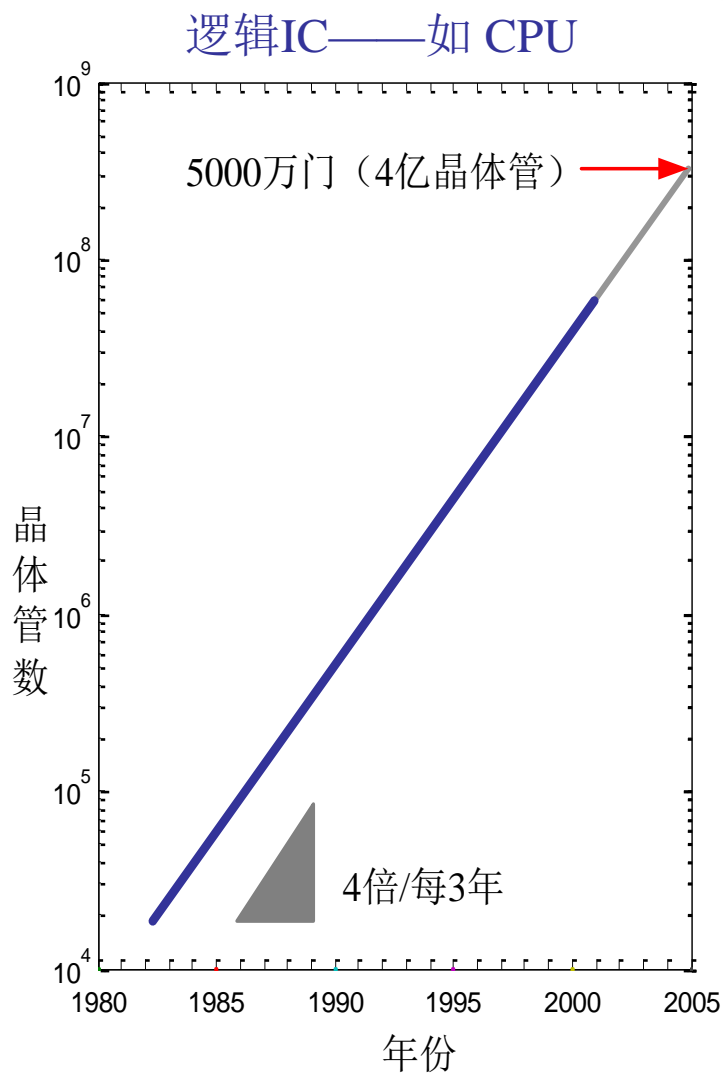


- ❑ EDA (Electronic design automation) 阶段
  - ❑ VLSI (Very-large-scale integration)
  - ❑ 集成的晶体管数目达到10万以上
  - ❑ SOC (System on a Chip)
- ❑ 需要进一步提高设计自动化水平
- ❑ 采用 HDL 设计
  - ❑ Hardware Descriptions Language (HDL)
  - ❑ ——硬件描述语言
- ❑ 特点
  - ❑ 用 HDL语言进行设计输入
  - ❑ 利用自动化综合软件, 自动生成电路及版图
  - ❑ 用仿真软件完成功能测试仿真



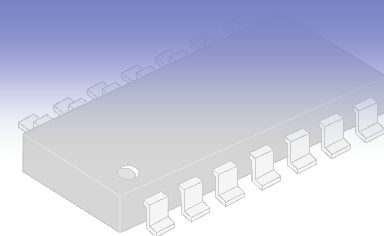
Intel Pentium (IV) microprocessor

# 数字集成电路复杂程度的发展趋势



Feature size: 晶体管或连线在 x 或 y 方向的最小尺寸

# 数字集成电路设计发展趋势

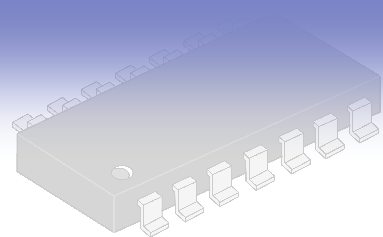


- ❑ 大规模专用集成电路（ASIC）
- ❑ 应用处理器（Application processor）
  - ❑ 包括多个 CPU 和 DSP IP 核（core）
    - ◆ DSP —— Digital Signal Processing（数字信号处理）
    - ◆ IP —— Intellectual Property（知识产权）
- ❑ 片上系统（SoC, System on chip）
- ❑ 片上可编程系统（SoPC, System on programming chip）



- (1) Process: TSMC 7nm FinFET
- (2) Octa-core CPU:
  - 2x Cortex-A76 @ 2.6GHz, (面向 Windows Laptop)
  - 2x Cortex-A76 @ 1.92GHz,
  - 4x Cortex-A55 @ 1.8GHz
- (3) GPU: Mali-G76 MP10
- (4) AI processing unit: 2x NPU (Neural processing unit)
- (5) RAM: LPDDR4X @ 2133MHz
- (6) LTE modem: LTE Cat 21, 1.4Gbps down / 200Mbps up

# 数字集成电路设计类型



## □ 两种设计类型

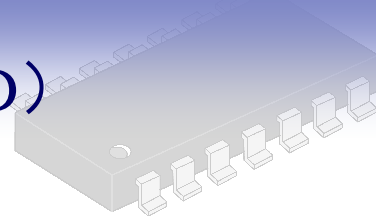
- ASIC 芯片设计
- 基于PLD的设计—— 基于FPGA/CPLD的设计

## □ ASIC (application specific integrated circuit)

- 把一个有专用目的，并具有一定规模的电路或子系统设计、集成在一个芯片上
- 设计方法
  - ◆ 全定制电路设计方法
    - 从设计基本单元开始，用搭积木的方式完成电路设计
  - ◆ 半定制电路设计方法
    - 采用标准的单元库进行设计， 用搭积木的方式完成电路设计
  - ◆ 主要问题
    - 设计开发周期长
    - 设计出的产品性价比与最终产品的数量相关



# 可编程逻辑器件（Programmable Logic Device, PLD）



## □ 1970年代发展起来的一种新型逻辑器件

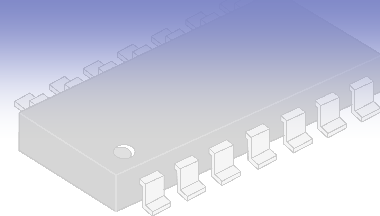
- 随着IC制造工艺的不断进步，PLD在性能和规模上不断发展
- VLSI 制造工艺导致高密度 PLD 的诞生，目前已达到 1000 万门以上的规模
- PLD器件不仅可以替代标准逻辑器件，也可以替代专用集成电路（ASIC）

# PLD设计和配置



- ❑ 使用EDA设计开发系统进行电路设计
- ❑ 定义和设计芯片内部的逻辑功能
  - ❑ 基于Verilog、VHDL等HDL
- ❑ 配置逻辑电路模块输入/输出之间逻辑关系（功能）
- ❑ 指定（或约束）输入/输出引脚
- ❑ 使用某种编程技术使内部电路结构实现再连接
  - ❑ ——电路的物理实现

# CPLD/FPGA



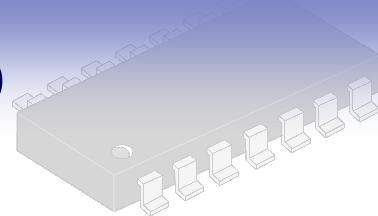
## □ 1980年代末

- Altera 公司推出了大规模的复杂可编程逻辑器件（CPLD）
  - ◆ Complex Programmable Logic Device
- Xilinx公司推出了超大规模的现场可编程逻辑门阵列器件（FPGA）
  - ◆ Field Programmable Gate Array
- CPLD/FPGA——具有一定连线的结构和已封装好的全功能的标准电路



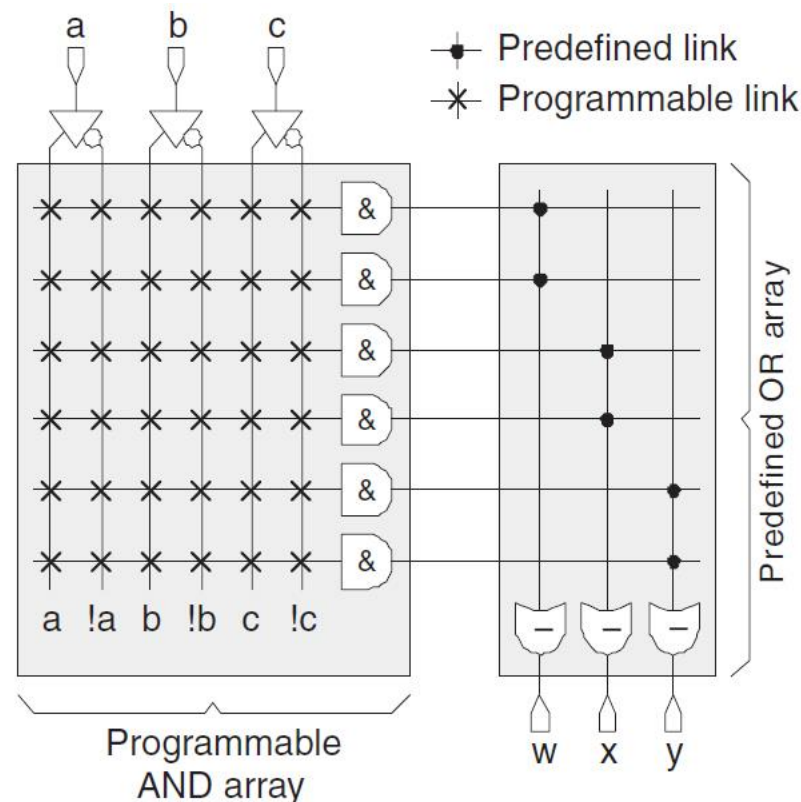
最昂贵的芯片：宇航级抗辐射FPGA

# CPLD (Complex Programmable Logic Device)

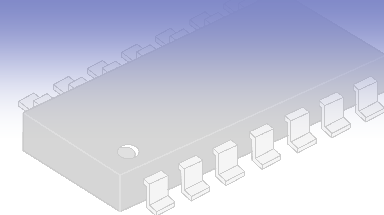


## ❑ 复杂可编程逻辑器件

- ❑ 从PAL和GAL器件发展出来的器件，一片就能实现数千、甚至数十万个逻辑门才能构成的电路
- ❑ PAL —— programmable array logic
- ❑ GAL —— generic array logic



# FPGA (Field Programmable Gate Array)

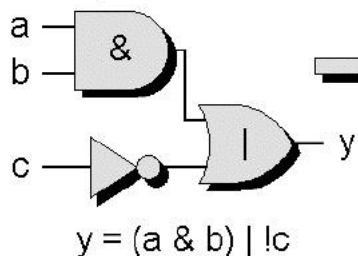


## 现场可编程逻辑门阵列

- 在PAL、GAL、CPLD等可编程逻辑器件的基础上发展起来的可编程逻辑器件
- 3-input LUT —— 3-input lookup table
- lookup table —— 查找表

$$y = (a \& b)!c$$

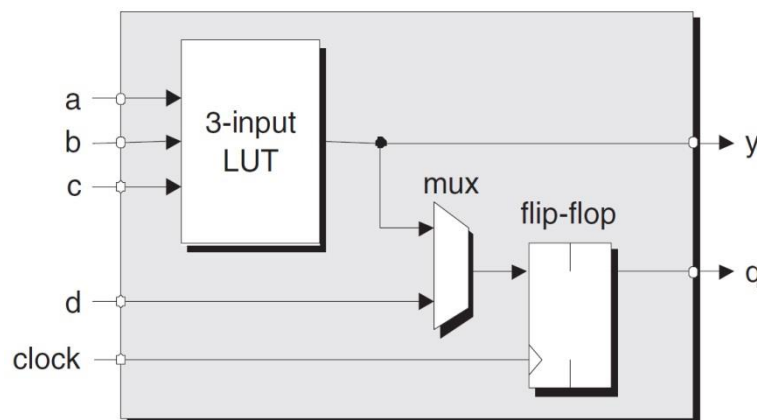
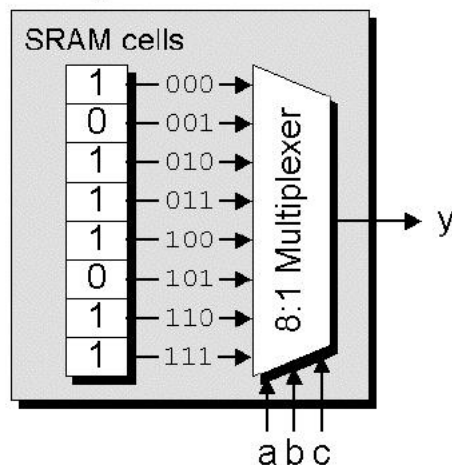
Required function



Truth table

a	b	c	y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Programmed LUT

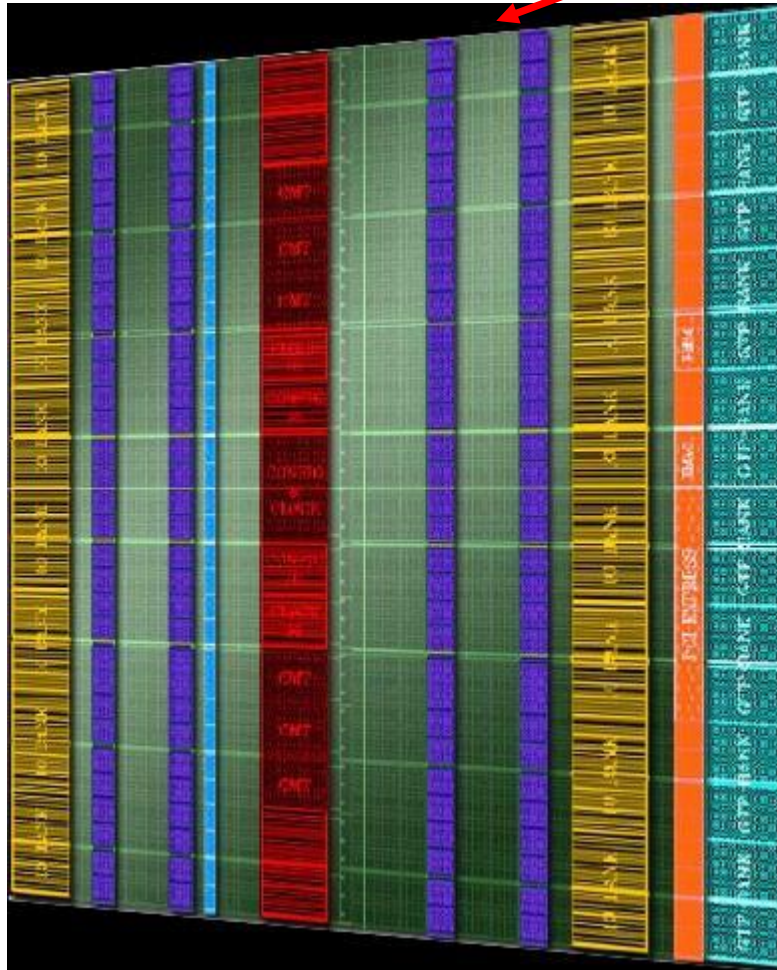
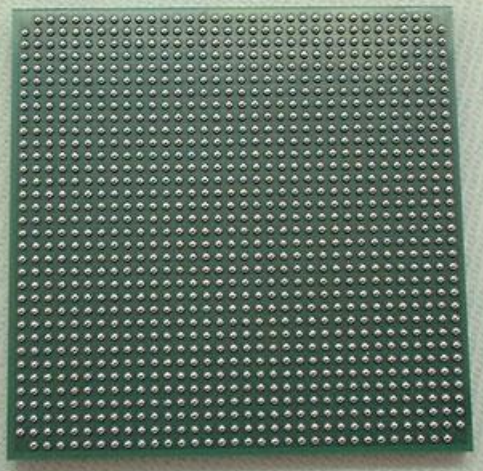
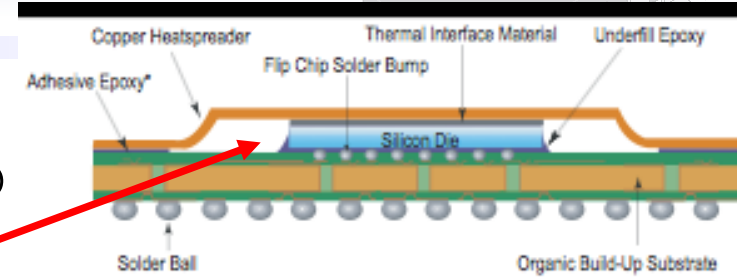




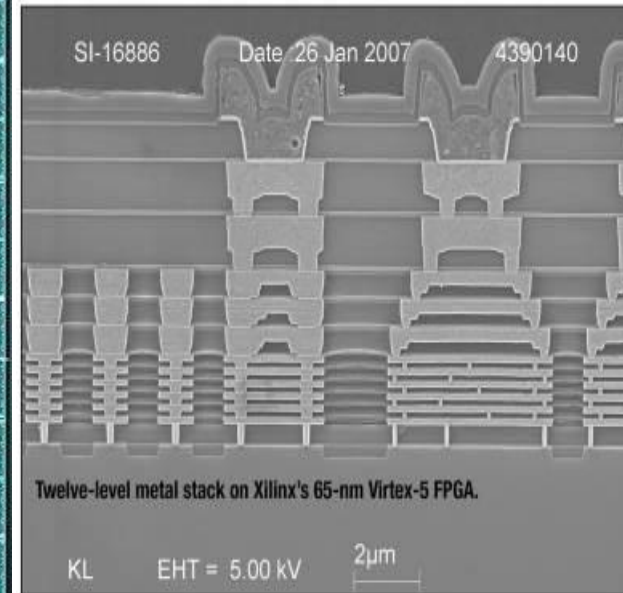
# FPGA——Xilinx Virtex-5 XC5VLX110T

## ❑ Virtex-5 die photo

❑ A die is an unpackaged part (没有封装的芯片)



Virtex-5:65nm, 12 metal layers



# CPLD/FPGA 特点



## □ 采用最先进的工艺

- CPLD/FPGA制造广泛采用深亚微米制造工艺

## □ 编程简便

- ◆ 可容易地实现红外线编程、超声编程或无线编程，或通过网络远程编程

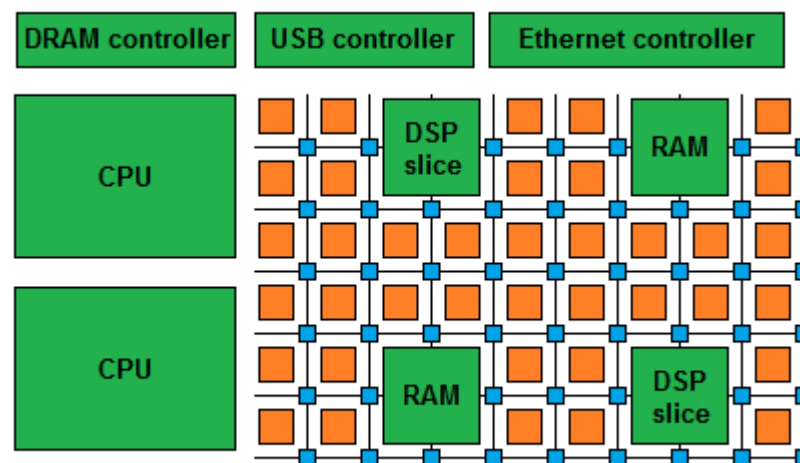
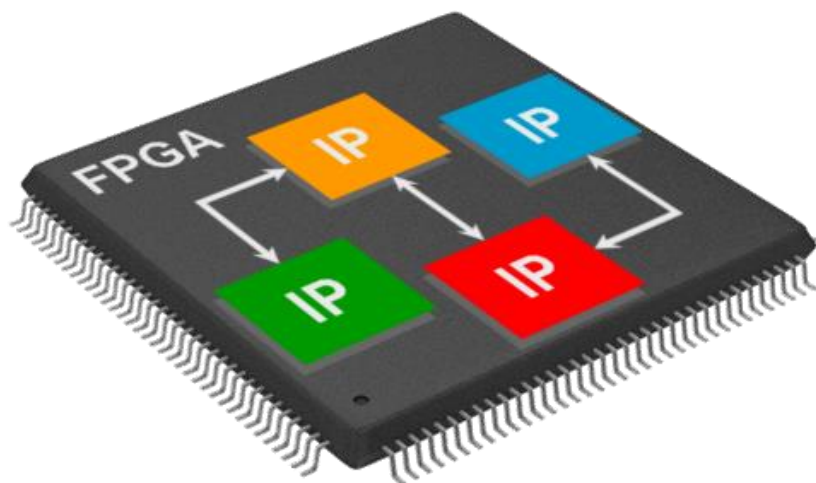
## □ 各种软/硬 IP 核

- CPLD/FPGA和IP核的结合极大的提高了系统性能

- ◆ IP—— Intellectual Property（知识产权）

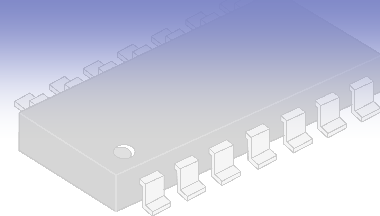
## □ CPLD/FPGA 在数字系统设计中已具有不可替代的地位

- 与 CPU、MCU、DSP 等器件间的功能和界限已日益模糊



Modern FPGA: lots of **hard**, not-field-programmable gates

# 基于FPGA的数字系统设计



## ❑ 采用EDA工具设计开发FPGA数字系统

- ❑ 开发周期短
- ❑ 设计系统易学易用，开发便捷

## ❑ FPGA的特点—— 具有硬件的处理速度，具有软件的灵活性

### ❑ 通信领域

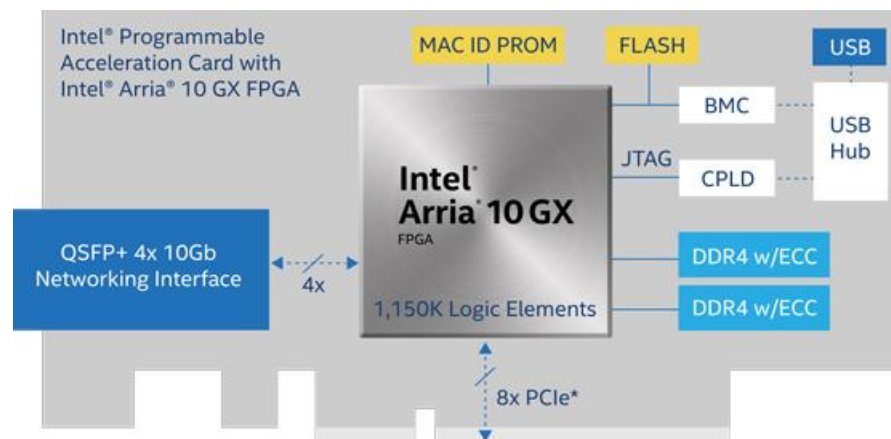
- ◆ 对速度要求比较高
- ◆ 数据处理比较复杂

### ❑ 信息安全领域

- ◆ 实现硬件加速包扫描
- ◆ 实现硬件加密/解密处理

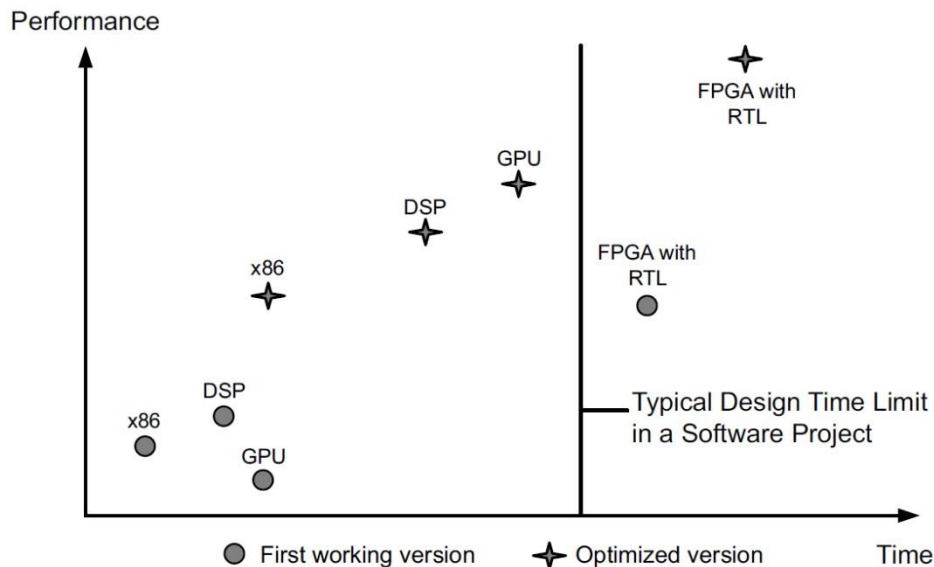
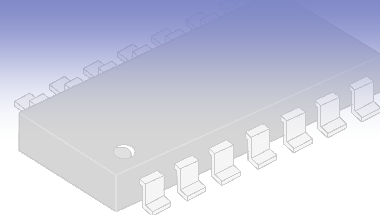
### ❑ 高性能计算

- ◆ 采用FPGA作为协处理器
- ◆ 通过对 FPGA 进行特定程序算法优化
- ◆ 可以大大提高对特定应用程序的执行效率

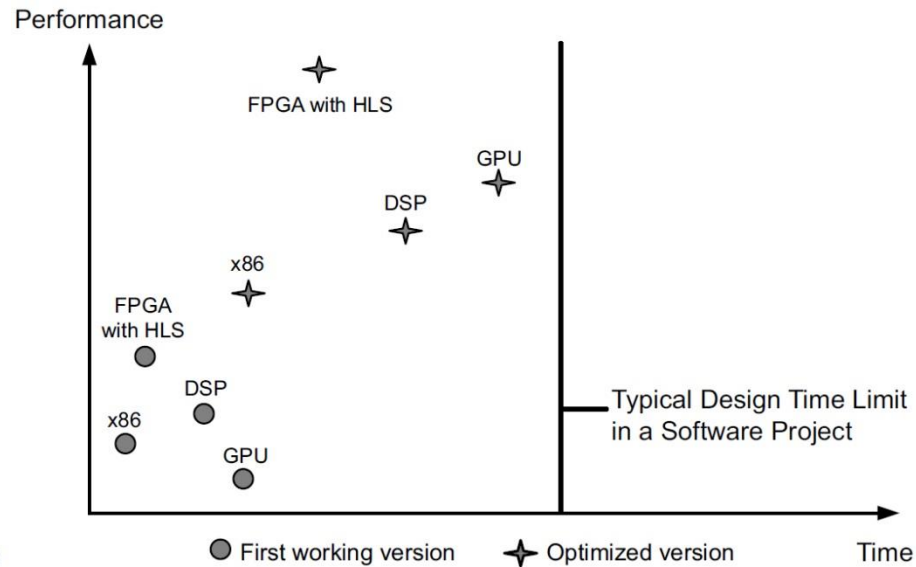


Intel Deep Learning Inference Accelerator

# 基于 FPGA 数字系统的性能与开发周期



- ❑ 使用RTL设计达到的性能和开发周期
- ❑ RTL—— Register transfer level



- ❑ 使用HLS 设计达到的性能和开发周期
- ❑ HLS —— High-Level Synthesis
  - ❑ 高层次综合
  - ❑ 使用 C/C++, SystemC