

第一章

数字系统设计概述



目录

1

数字系统概念

2

数字系统的实现方法

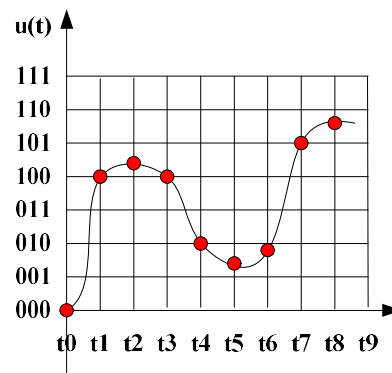
3

基于HDL和EDA工具的设计流程

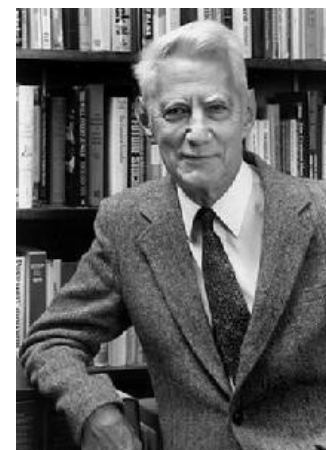


数字信号

- 现实中大部分信号是模拟的（连续的）
- 模拟信号数字化处理：采样、量化
 - 转换：数模、模数
- 模拟信号与数字信号
 - 稳定性
 - 数学运算
- 模拟电路与数字电路
 - 结果的再现性
 - 易于设计
 - 功能性

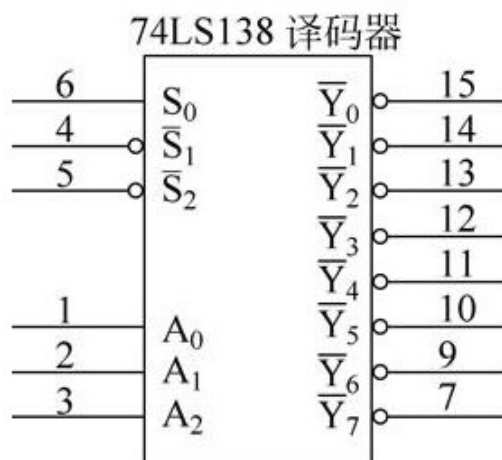


000 001 010 011 100 101 110 111
编码



□ 功能级电路

- 

[illegible]

数字系统的概念

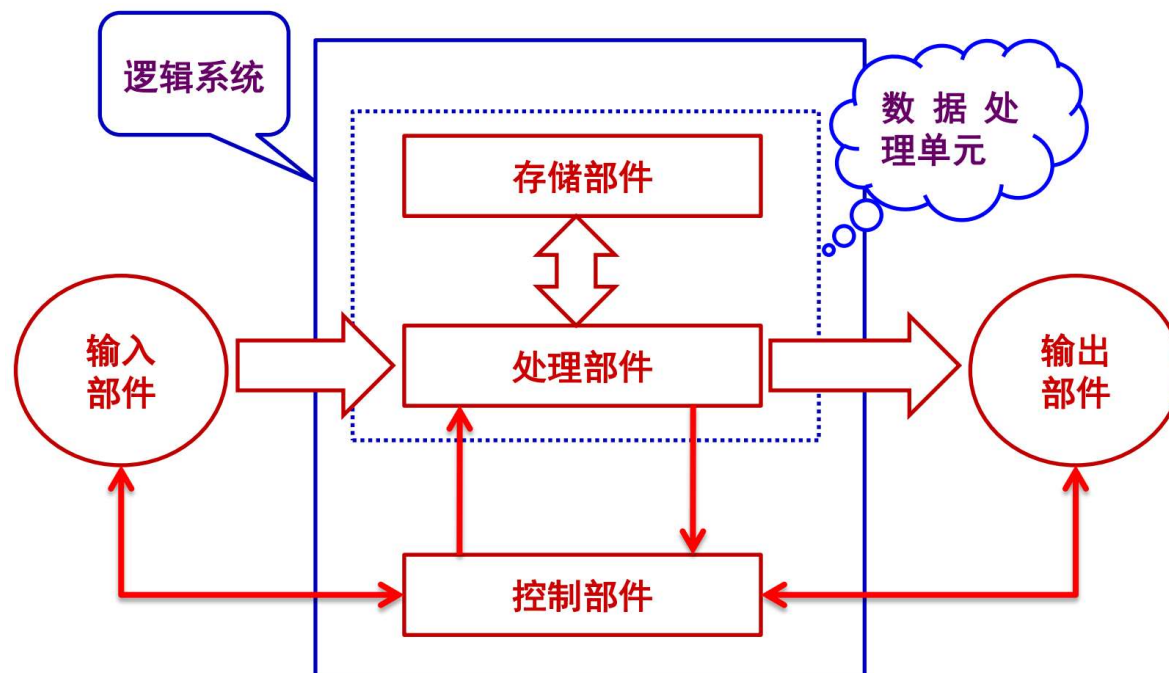
□ 数字系统 (Digital System)

- 由若干数字电路和逻辑部件构成的、能够实现**数据存储、传送和处理等复杂功能**的数字电子系统
 - 通用处理器：CPU
 - 专用数字系统：GPU、存储器、调制解调器等



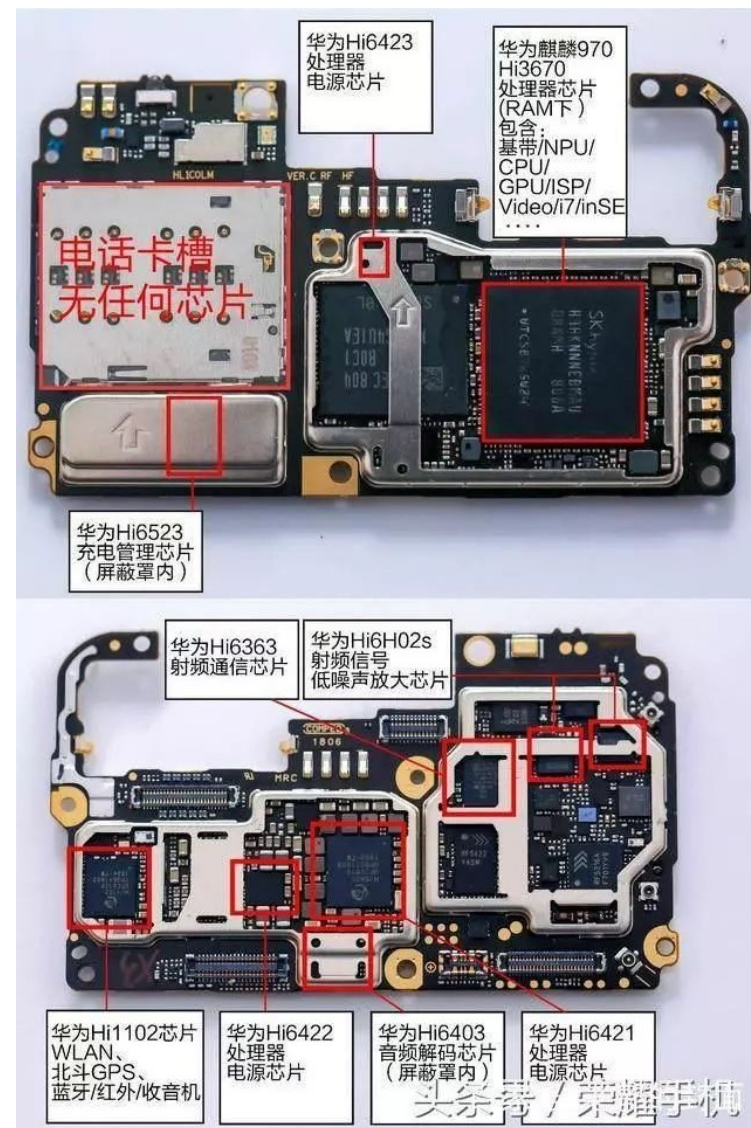
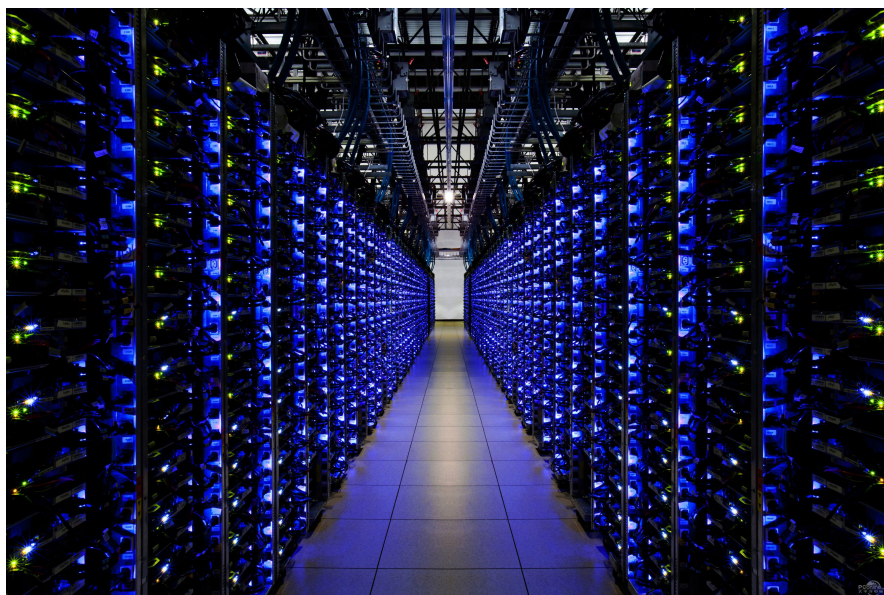
数字系统的概念

- 数字系统在结构上分为：处理器 + 外围接口
- 处理器具体可分为：数据处理单元 + 控制单元

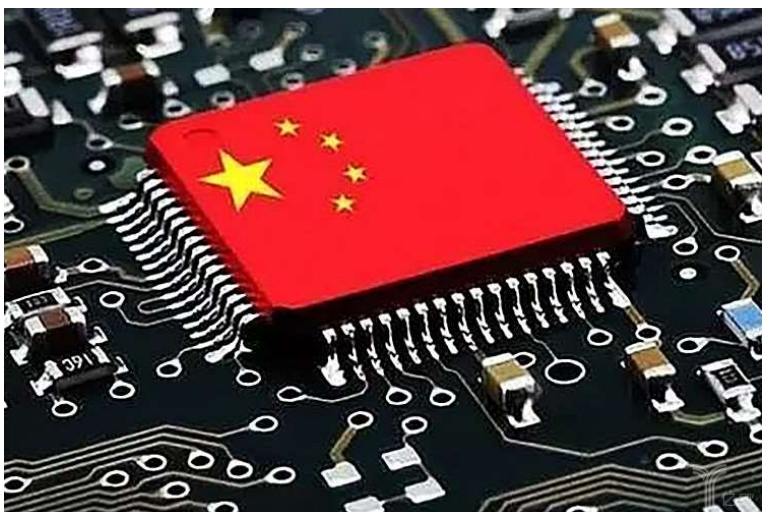


数字系统内部所传输处理的对象均为基本数字逻辑变量

数字系统——无处不在



中兴事件+华为事件



- 核心技术和关键设备受制于人，中高端集成电路依赖进口
- 人才严重不足，制约我国集成电路产业发展



目录

1

数字系统概念

2

数字系统的实现方法

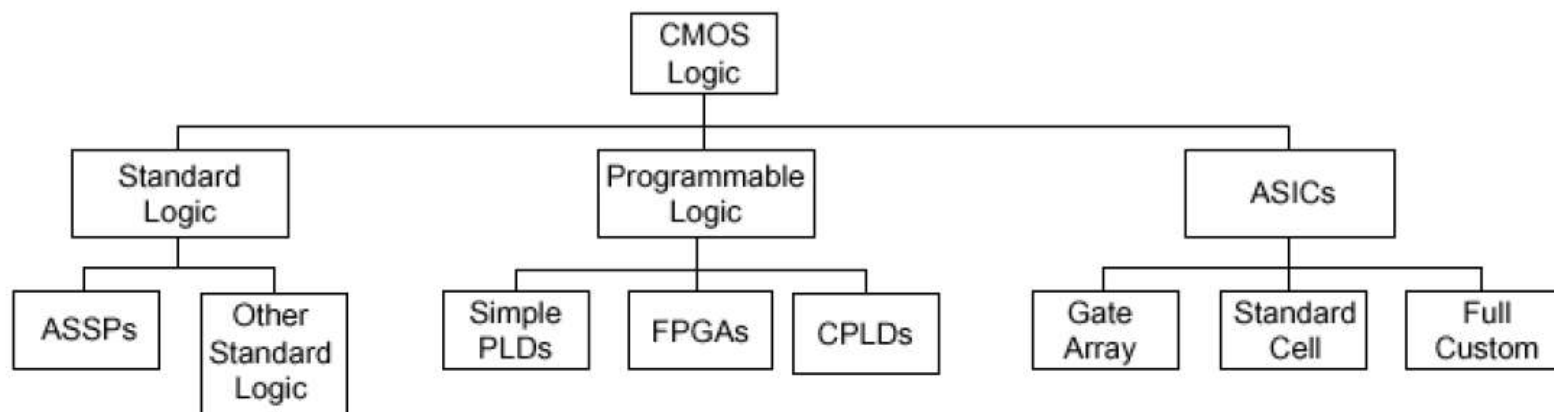
3

基于HDL和EDA工具的设计流程



数字系统的实现方法

- ASSP (Application Specific Standard Product) 专用功能标准产品
- PLD (Programmable Logic Device) 可编程逻辑器件
- ASIC (Application Specific Integrated Circuit) 专用集成电路



数字系统的实现方法-ASSP

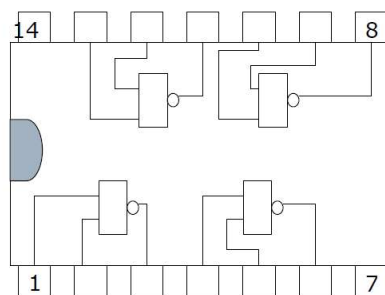
□ 专用功能标准产品

- 具备特定功能的芯片：74系列、4000系列
- 反相器、或门、或非门、与或非门、比较器、译码器

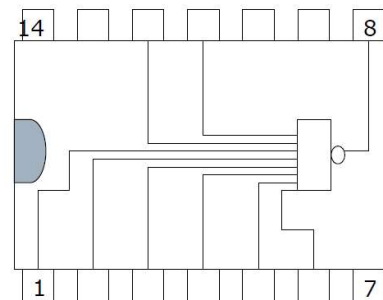


□ 特点

- 非常通用
- 功能相对简单
- 适合设计小规模系统



74LS00
2输入4与非门

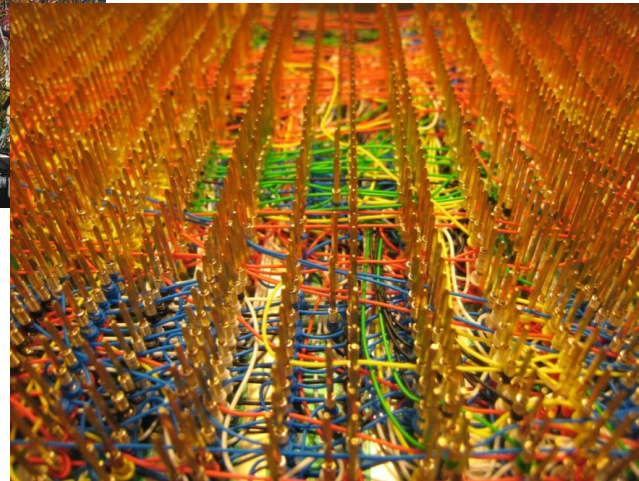
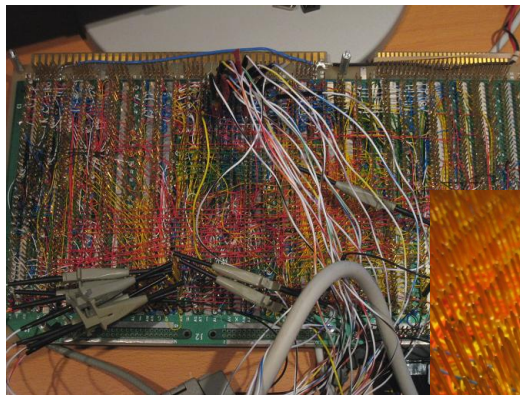


74LS30
8输入与非门

数字系统的实现方法-ASSP

□ 纯手工搭建CPU

- 2009年世界创意大会, Steve
- Big Mass Of Wires



数字系统的实现方法-PLD

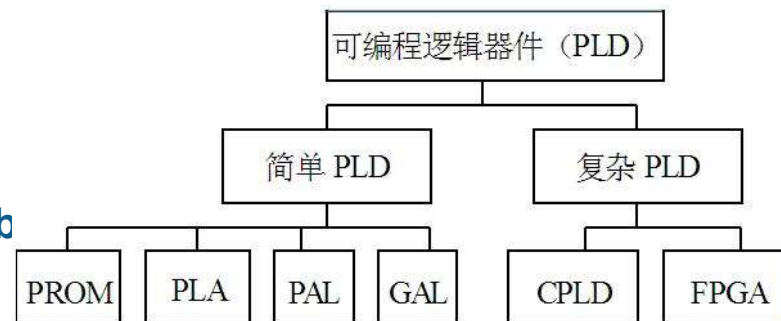
□ 可编程逻辑器件

□ 分类

- 复杂可编程逻辑器件(Complex Programmable Logic Device, CPLD) – 小规模
- 现场可编程门阵列 (Field - Programmable Gate Array, FPGA) – 大型系统

□ 特点

- 灵活
- 规模：几百-上千万门
- 价格：几十-几万人民币
- FPGA接口丰富
- 设计开发周期短、设计制造成本低
- 适用于非批量生产应用



数字系统的实现方法-ASIC

□ 专用集成电路

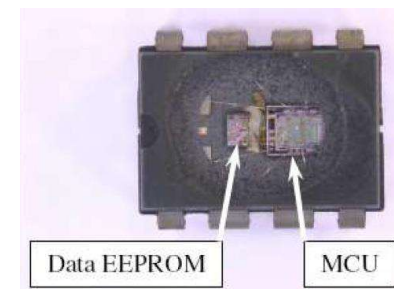
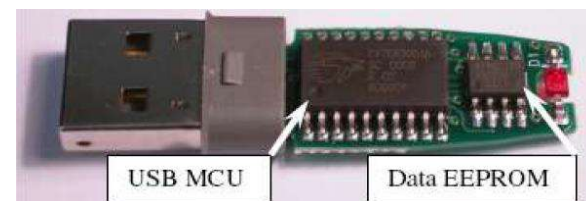
- 针对某一电路系统的要求专门设计
- 具有特定电路功能，通常市场上买不到
- 性价比好
 - 体积小、低功耗
 - 可靠性
- 应用广泛



集成电路产业特点

- 集成电路产品周期比较短
 - 电子产品更新换代非常快
 - 先进的设计方法学

- 设计与制造分开
 - Fabless公司
 - 加工、封装、测试
 - 联发科





目录

1

数字系统概念

2

数字系统的实现方法

3

基于HDL和EDA工具的设计流程



传统的设计方法-以组合逻辑电路设计为例

1. 确定输入信号和输出信号
2. 确定输入和输出的逻辑关系
 - 逻辑函数
 - 真值表
3. 优化逻辑表达式
 - 卡诺图化简等
4. 实现电路

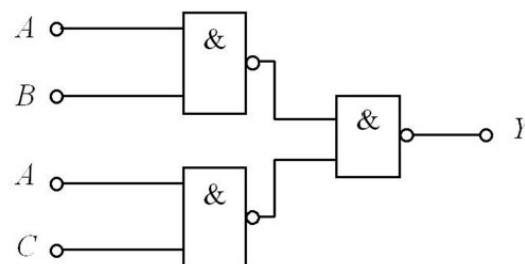
$$y1 = a \text{ and } b \text{ or } c$$

$$y2 = d + e$$

输入			输出
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Y	BC				
	A	00	01	11	10
0					
1		1	1	1	

$$Y = AB + AC = \overline{\overline{AB}} + \overline{\overline{AC}} = \overline{\overline{AB} \cdot \overline{AC}}$$



第3、4部可以
交给计算机自
动完成

新的组合逻辑电路的设计步骤

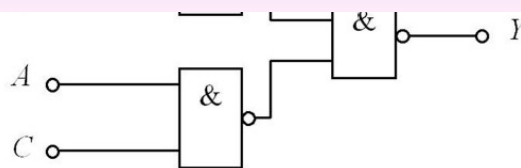
1. 确定输入信号和输出信号
2. 确定输入和输出的逻辑关系
3. 用HDL（硬件描述语言）正确描述电路功能

输入			输出
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

```
module aa(y,a,b,c)
```

让计算机软件优化并生成电路

```
assign y = a&b+a&c;  
endmodule
```





基于HDL的数字系统设计方法

□ 关键要素

- 数字系统设计EDA工具
- Top-down设计思想
- 硬件描述语言 (HDL)



Bottom-up设计

□ 流程

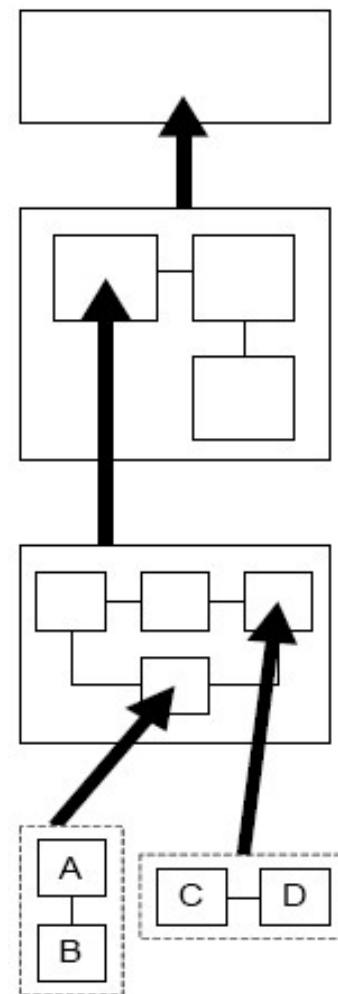
- 首先决定使用的器件（如74系列器件、RAM、ROM、CPU等）
- 然后构成各个功能模块（如数据采集、处理、通信接口等模块）
- 最后合成整个系统。

□ 问题

- 在进行底层设计时缺乏对系统总体性能的把握，在合成整个系统后，可能发现无法达到要求的功能或性能，**难以保证一次设计成功率。**

□ 结果

- 效率低，不能满足大规模系统的设计需要。



Top-down基本设计流程

□ 系统设计

- 需求定义、软硬件设计划分
- 算法设计、体系结构设计

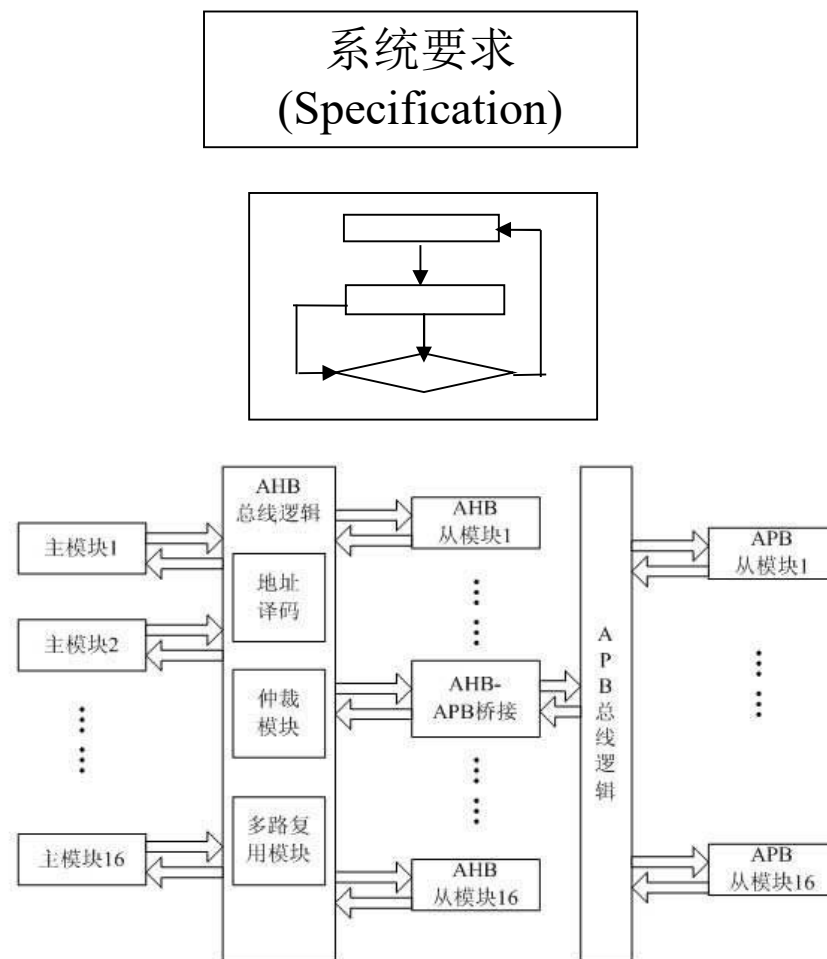
□ 逻辑设计

□ 仿真、综合及优化

□ 物理设计

- 布局规划、布局、布线

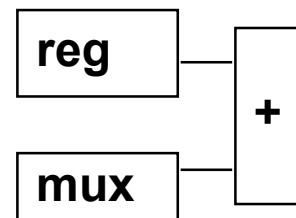
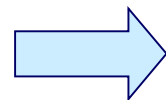
- **优点：**在系统级完成整个系统的模拟和性能分析，自顶向下进行每层次的设计和仿真验证，保证最后设计结果的正确性，**设计成功率高**



逻辑设计

- 主要通过**硬件描述语言**的行为描述实现
- 寄存器传输级 (RTL: Register Transfer Level)
 - 描述数据在寄存器之间流动和如何处理这些数据的模型
 - 描述的是每一个clock下，触发器集合A的旧值进行逻辑运算，产生新值，存入触发器集合B的过程。
 - 可综合得到电路实现

```
if (a == b)
    out = 1;
else
    out = 0;
```



- 逻辑综合，之后的设计基本由EDA工具自动完成

什么是硬件描述语言HDL

- 具有特殊结构能够对硬件逻辑电路的功能进行描述的一种高级编程语言
- 这种特殊结构能够：
 - 描述电路的连接
 - 描述电路的功能
 - 在不同抽象级上描述电路
 - 描述电路的时序
 - 表达具有并行性
- HDL语言：Verilog、VHDL、system Verilog
 - 都是IEEE标准
 - 都被ASIC和FPGA综合工具支持
 - 近年chisel

Verilog的历史

- 1983年, **GDA**(GateWay Design Automation)公司的**Phil Moorby**创立**Verilog HDL**。Phil Moorby后来成为Verilog-XL的主要设计者和**Cadence**公司的第一个合伙人
- 1990年, Cadence公司收购了GDA公司
- 1991年, Cadence公司公开发表Verilog语言, 成立了OVI(Open Verilog International)组织来负责Verilog HDL语言的发展
- 1995年制定了**Verilog HDL的IEEE标准, 即IEEE1364**
- 2001年, Verilog标准进一步完善, 形成1364-2001即目前主流版本
- 2005年, 出现修正版本**1364-2005**
- 2009年, IEEE标准1800-2009, 也称为System Verilog
- 2012年, IEEE 1800-2012, 最新版本



Verilog的特点

- 描述电路的行为
- 依靠EDA工具综合出具体的电路
- 优点
 - 工艺无关性
 - 可移植性
 - 易于维护



Verilog的抽象层次

- Verilog模型可以是实际电路的不同级别的抽象。这些抽象的级别和它们对应的模型类型共有以下五种
 - 系统级(system): 用高级语言结构实现设计模块的外部性能模型
 - 算法级(algorithmic): 用高级语言结构实现设计算法的模型
 - **RTL级(Register Transfer Level): 描述数据在寄存器之间流动和如何处理这些数据的模型**
 - 门级(gate-level): 描述逻辑门以及逻辑门之间的连接的模型
 - 开关级(switch-level): 描述器件中三极管和储存节点及其之间连接的模型