



# 第七讲 SOC系统测试与分析

- SoC系统验证方法
- SoC系统测试原理及方法





## 7.1 SoC系统验证方法

在系统芯片的设计过程中，系统规约确定之后进行系统级设计。首先对系统行为进行建模，根据功能规范要求对行为模型进行验证；然后将行为模型映射到由芯核和功能块组成的架构之上。目的就是去验证该架构的功能和性能。





## 7.1 SoC系统验证方法

在功能设计和架构映射之后，都需要进行验证，分别是功能验证和性能验证。

**功能验证：**

目的是检查行为设计是否满足功能需求。

**性能验证：**

目的是检查所选出的架构是在满足功能需求之外是否能满足性能需求。





## 7.1 SoC系统验证方法

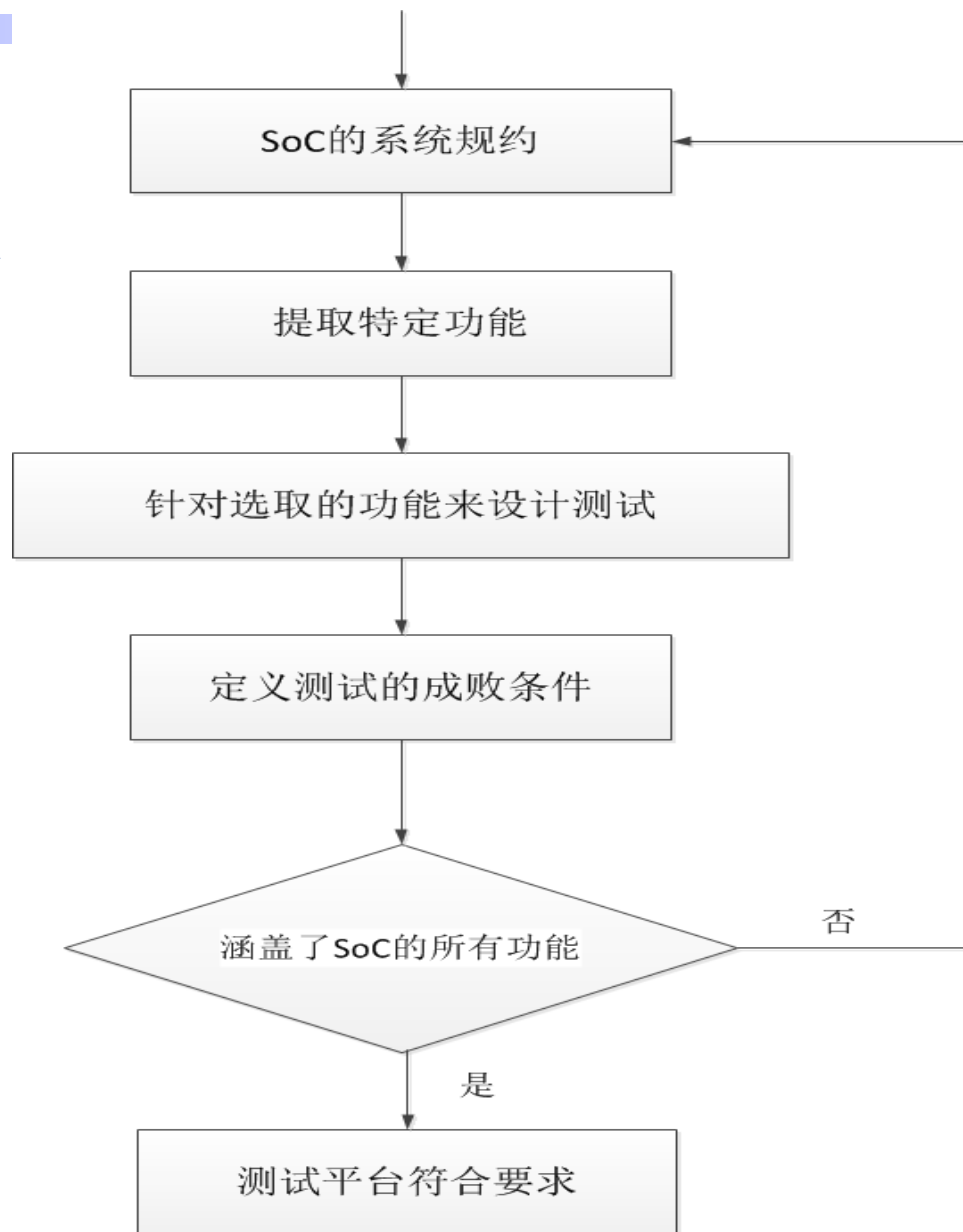
在整个验证过程中，都将使用测试平台来检验设计对象的功能，系统级测试平台是整个验证过程的一个关键。





## 7.1 SoC系统验证方法

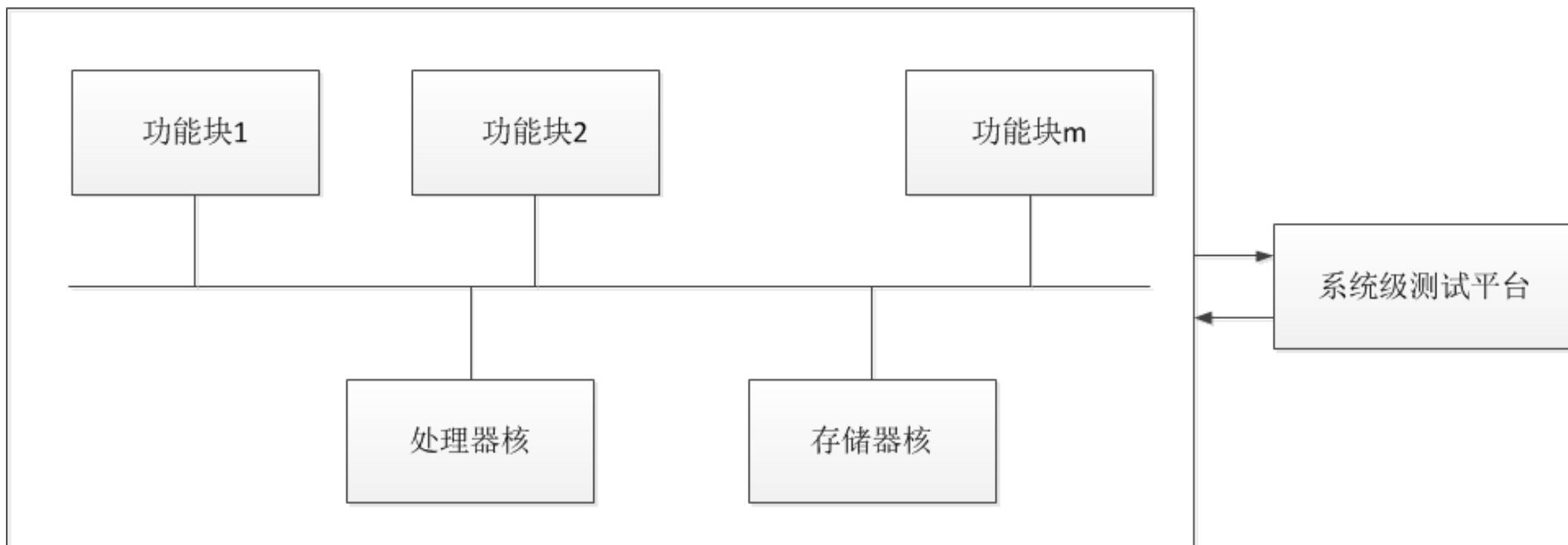
从系统规约中提取出一项功能要求，并定义出检验其功能的具体测试，重复进行，直至为每一项功能都建立了测试。





## 7.1 SoC系统验证方法

- 在实际中对SoC进行验证时，由于它是由多个功能块组成，可以将SoC的整个系统级测试平台运用于系统芯片的每一个子模块（功能块），实现对每个功能块的细节进行验证。





# 7.1 SoC系统验证方法

- 对SoC功能块的细节进行验证时，可以采用如下多种方法：硬件建模、接口验证、软/硬件协同验证、随机测试、基于应用程序的验证、门级验证等。
- 硬件建模
  - 为了达到足够高的无故障率，需要运行大量的应用程序来进行测试。
  - 对于规模较小的设计，可以使用单个FPGA对整个芯片进行建模
  - 对于规模较大的设计，可以使用多个FPGA进行建模，配合专门开发的软件，构建硬件仿真器





## 7.1.1 硬件仿真器

硬件仿真器提供了对可重配置逻辑、可编程互连、大容量逻辑以及特殊存储器和处理器的支持。

对**SoC**的设计，若要使用硬件仿真，就需要提供适合于硬件仿真器的各种模型，如微处理器、存储器、总线功能模型、监控器、时序产生器等。这些模型的建立应该在**SoC**设计阶段的早期就开始进行，可以与芯核的提供商一起来共同解决。







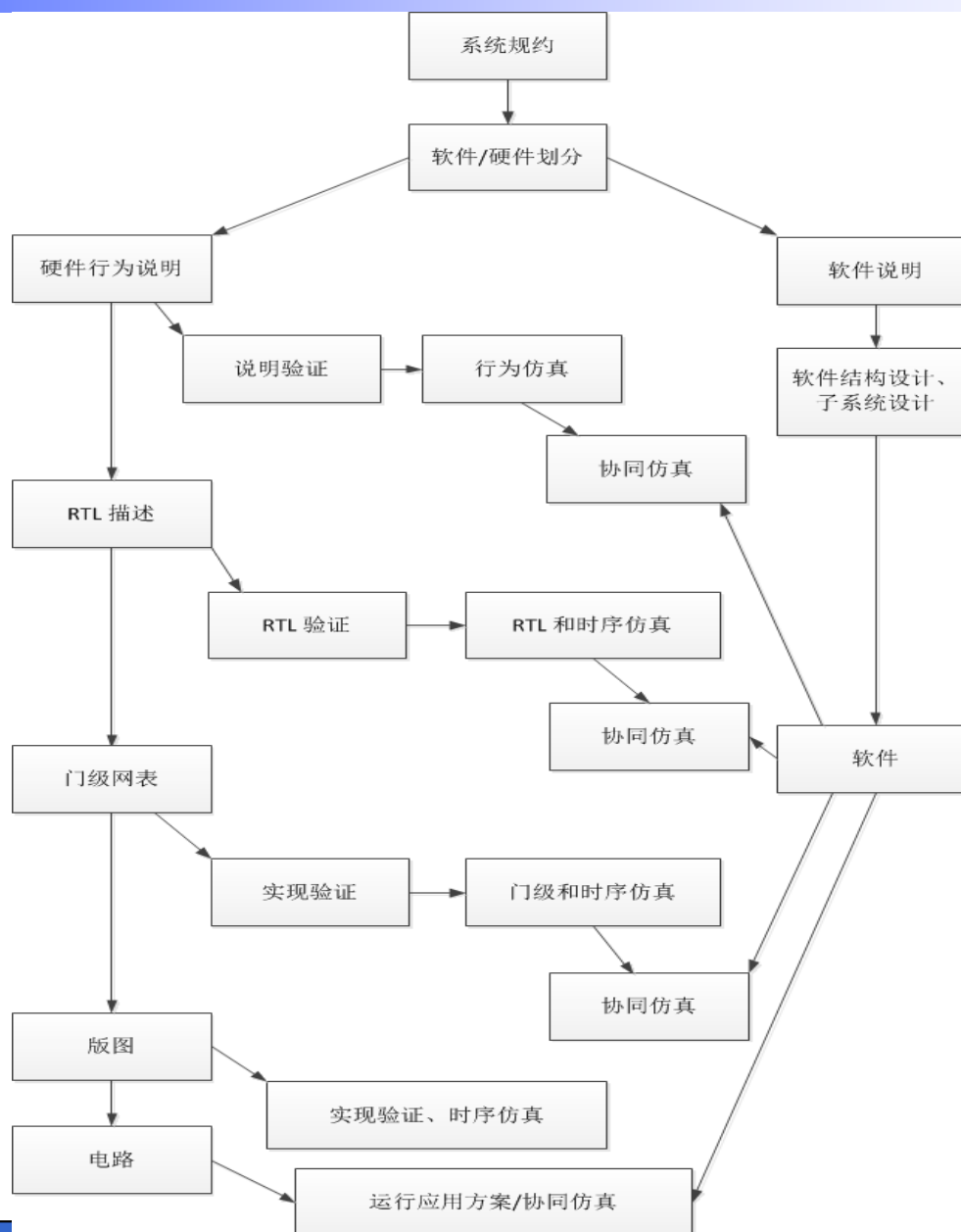
## 7.1.2 协同验证与仿真

- 在设计阶段的早期建立协同验证环境，进行软件/硬件协同验证。使用协同验证可以达到如下目标：让电路工程师们能够同时设计、开发和调试软件和硬件，同时在系统集成和制造芯片之前将设计错误去除。





## 7.1.3 协同验证与仿真





## 7.1.3 协同验证与仿真

从理论上讲，在协同仿真中，硬件可以用C/C++建模，整个系统可以像单个C/C++程序一样执行。实际上，对硬件的实现，仍是采用HDL/RTL描述。因此，协同仿真需要一个或多个HDL仿真器和一个C/C++平台（编译器、装入程序、链接器和计算机操作系统的其他部分）。





## 7.1.4 硬件仿真

- 在协同仿真中，包括HDL仿真器和软件仿真器在内的两个或多个仿真器需要互相链接，因此不同仿真器之间的通信是关键问题，可以采用主/从模式、分布式模式两种方案。





## 7.1.5 系统级时序验证

- 时序验证是用来检查设计对象是否满足预期的时序要求。对所设计电路中的每一个存储单元和锁存器都存在需要满足的时序要求，如建立时间、保持时间、延迟时间等。
- 时序分析有动态分析和静态分析两种类型。
- 动态时序分析使用仿真向量去验证；对于给定的输入，电路的输出结果是否符合时序的规定。
- 静态时序分析检查电路中的所有时序路径





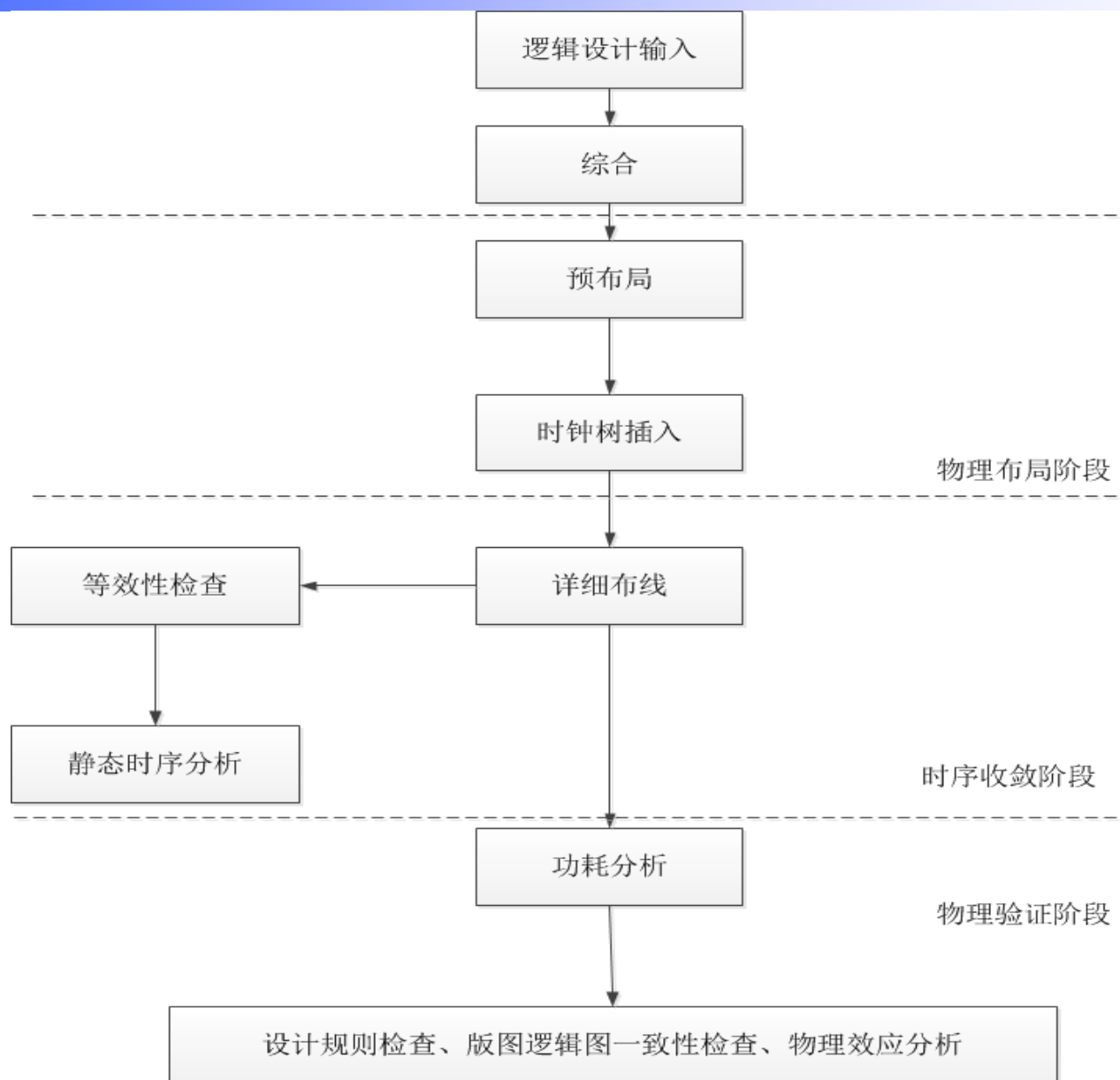
## 7.1.6 物理验证

- 物理设计是在逻辑设计或电路设计之后实现物理版图的过程。
- 物理版图是由各层次的版图编辑而成，形成晶体管和其他器件相互间的连线。
- 物理验证是物理设计的一个重要环节。





## 7.1.6 物理验证





## 7.1.6 物理验证

- 物理验证阶段的任务是对最后结果进行验证，主要有功耗分析、设计规则检查、版图逻辑图一致性检查、物理效应分析等工作。



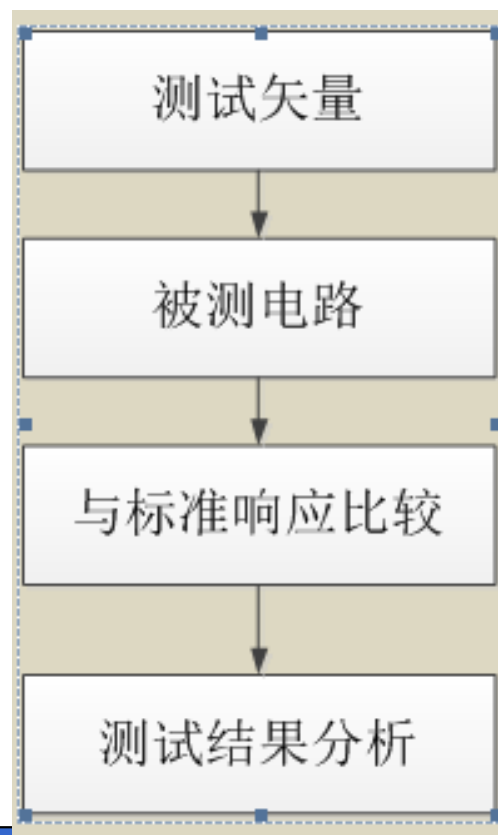




## 7.2 SoC系统测试原理及方法

现在人们对电路的测试通常是在测试系统所提供的硬件、软件环境下实现的。

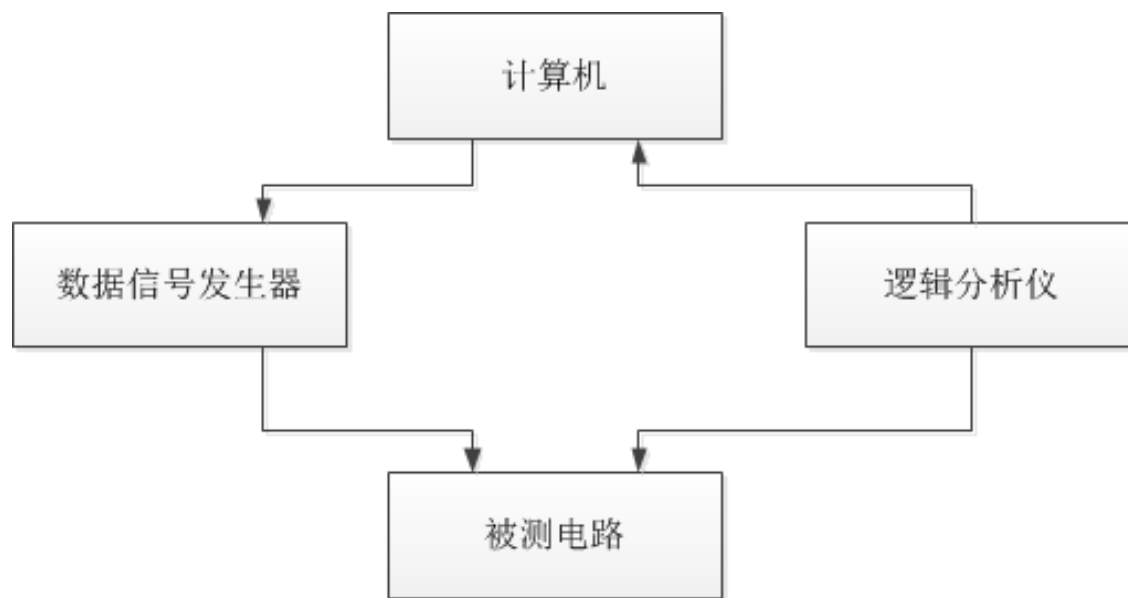
测试矢量是影响测试效果的关键，它可通过测试生成（**ATPG**）算法获得。





## 7.2 SoC系统测试原理及方法

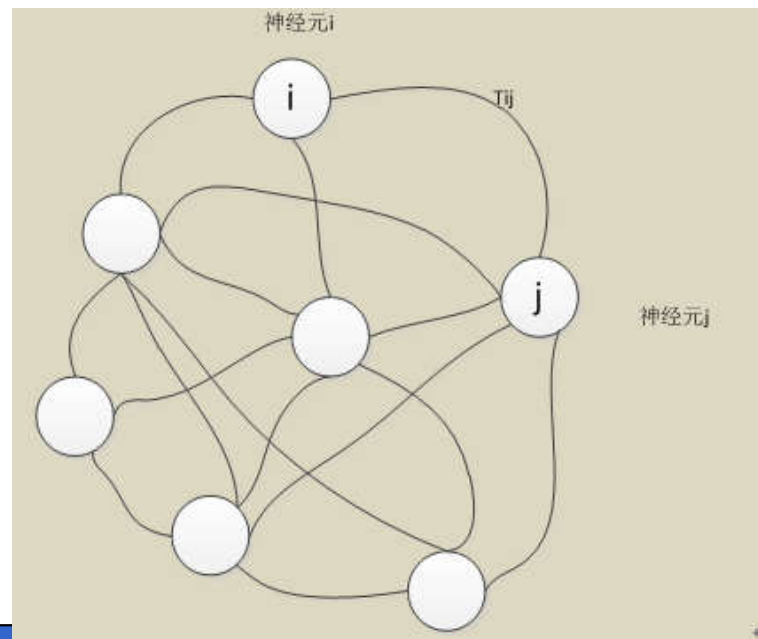
计算机主要用于为被测电路生成测试矢量；数据信号发生器根据计算机的要求产生测试波形，并加载到被测电路上；逻辑分析仪采集被测电路的响应信号并进行一定的分析，然后将结果送到计算机中进行处理。





## 7.2.1 基于神经网络的电路测试生成方法

- 人工神经网络（ANN）由于其优良的特性，能较好的处理目前串行计算机难于解决的NP完全问题（如Hopfield神经网络用于TSP问题的求解）。
- 根据组合电路测试生成的特点，选用Hopfield神经网络作为电路建模的基础，用神经网络的能量函数来表征电路的逻辑特性。





## 7.2.2 二元判定图BDD

- 二元判定图(BDD)就是一种较有效的方法，它将布尔函数的功能用有向无环图来表示，图中从根节点到叶节点的路径对应了布尔函数值为1的一个输入矢量。





## 7.2.3 大规模集成电路与系统的可测性设计

- 电路的可测性一般定义为测试的简便性或经济而有效测试的能力，主要涉及如下三个基本方面：测试矢量的产生、测试的评估和计算、测试的施加。
- 电路可测性设计的基础是可测性的度量方法





## 7.2.4 VLSI与系统的可测性设计

- 边界扫描可测性设计
- 数摸混合电路的边界扫描测试
- 专用集成电路设计中的边界扫描





## 7.2.4 VLSI与系统的可测性设计

- 边界扫描可测性设计

边界扫描测试是一种可应用于数字器件的具有结构化特性的测试技术，它通过在集成电路的设计阶段采取一些措施，能在很大程度上简化印制电路板级的测试，从而降低测试费用。







## 7.2.4 VLSI与系统的可测性设计

- 数摸混合电路的边界扫描测试
- 通过数模或模数转换器(ADC或DAC)将模拟电路的模拟信号转换为数字信号，并由边界扫描链输出，或者把测试模拟电路的模拟信号先将其数字信号形式由边界扫描管脚输入，加到相应的模数转换器上，完成对模拟器件的测试。
- 专用集成电路设计中的边界扫描
- 对规模较小的FPGA器件，若它未含有边界扫描测试端口，则进行可测性设计时应为其设计边界扫描测试端口，并连入边界扫描链。





# JTAG接口

JTAG(Joint Test Action Group 联合测试行动小组)

**JTAG**大致分两类，  
一类用于测试芯片的电气特性，检测芯片是否有问题；  
一类用于**Debug**

- **TCK**为
- **TDI**为
- **TDO**为测试数据输出
- **TMS**为测
- **/TRST**为

一个含有**JTAG** Debug接口模块的CPU，只要时钟正常，就可以通过**JTAG**接口访问CPU的内部寄存器和挂在CPU总线上的设备，如RAM，FLASH，GPIO等

效