

Verilog HDL

第一讲

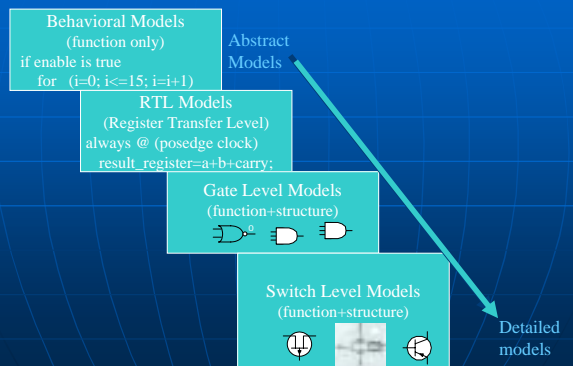
第一章 电子系统设计与硬件描述语言

- EDA设计主要流程
- 深亚微米技术给芯片设计带来的挑战
- 硬件描述语言

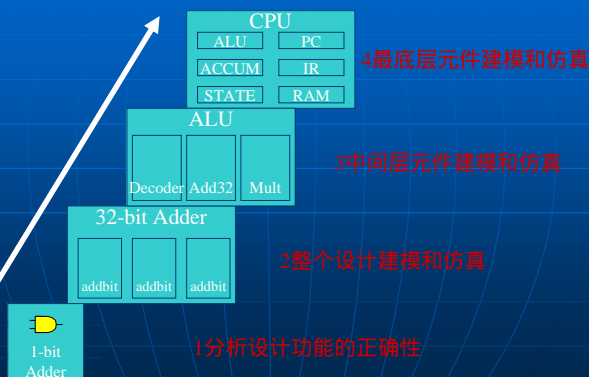
一、EDA设计主要流程

- 系统级模型
- 行为级描述
- RTL级描述
- 逻辑综合与优化
- 门级仿真与测试
- 布局布线
- 参数提取与后仿真
- TapeOut

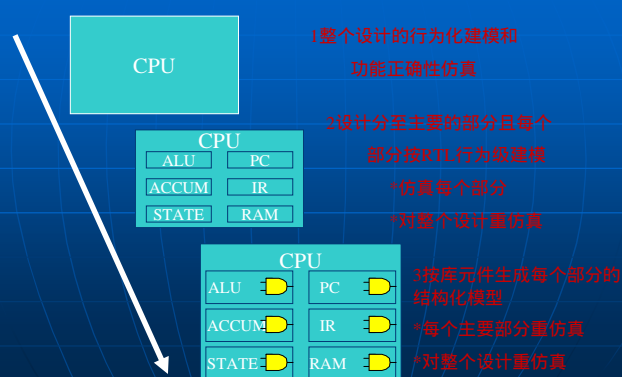
Verilog HDL中的描述层次



自底向上的设计流程



自顶向下的设计流程



二、深亚微米技术给芯片设计带来的挑战

最小特征尺寸

- < 1um 亚微米
- < 0.5um 深亚微米
- < 0.25um 超深亚微米

1. 连线延时比例变化

- 信号传输延时
= 门(器件)延时 + 连线延时
- 0.6um时: 70%门延时 + 30%门延时
- 0.5um时: 50%门延时 + 50%门延时
- 0.25um时: 70%门延时 + 30%门延时
- 版图布线的随机性直接决定了时序设计的难度

2. 电路功耗、系统时钟和可靠性的设计困难

- 功耗问题：
 - 器件密度提高导致单位面积的功耗上升
 - 时钟频率的提高导致器件电流增加
- 时钟（树）的同步设计带来困难
 - 连线延时增加
- 串扰和噪声影响可靠性
 - 线密度和器件密度增加
 - 工作频率的提高

3. 需要更精确的设计模型

- 更精确的器件模型
 - 模型参数的精确化
 - 模型参数的多样化
 - 分布参数模型
- 更精确的连线模型
 - 线负载模型 (Wire Load Model)
 - 分布参数
 - 设计过程中的统计模型

EDA技术的发展1

- 整体布局规划设计 (FloorPlan)
 - 逻辑设计和版图规划综合考虑
 - 模块划分和版图布局
 - 最少的模块连线
 - 最短的连线延时

EDA技术的发展2

- 低功耗设计和专用时钟布图
 - 降低工作电压：3.0V - 2.5V - 1.8V - 0.9V
 - 门控时钟
 - 门级低功耗优化设计技术
 - 时钟树布图设计工具，确保同步

EDA技术的发展3

- Cycle - Based驱动仿真技术
 - 由事件驱动变换到有效时钟驱动

第二章 Verilog HDL设计入门

- 一、BottomUp与TopDown的设计方法
- 二、VerilogHDL描述过程
- 三、硬件设计实现

一、BottomUp与TopDown的设计方法

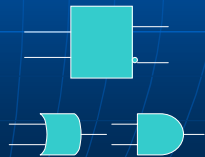
■ 1. BottomUp的设计方法

4. 完成整个系统的测试与性能耐分析

3. 由各个功能模块连成一个完整的系统

2. 由逻辑单元组成各个独立的功能模块

1. 由基本门构成各个组合与时序逻辑



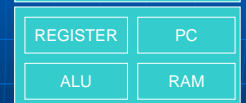
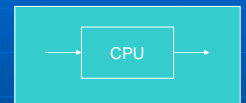
2.TopDown的设计方法

1. 系统层：顶层模块，行为级描述，功能模拟和性能评估

2. 各个功能模块划分，设计和验证

3. 各功能模块的系统级联合验证

4. 工艺库映射



2.TopDown设计方法的特点

- 从系统层开始设计和优化，保证了设计结果的正确性
- 适合复杂的、大规模电路的设计
- 缩短设计周期
- 依赖于先进的EDA工具和环境，费用昂贵
- 需要精确的工艺库支持

二、VerilogHDL描述过程

1. 与C语言的比较

VerilogHDL是在C语言基础上发展起来的，保留了C语言的结构特点。

- C语言由函数组成，VerilogHDL由模块（module）组成
- C语言通过函数名及其端口变量实现调用，VerilogHDL也通过模块名和端口变量实现调用
- C语言有主函数main（），VerilogHDL的各Module均等价，但必有一个顶层模块，包含芯片系统与外界的所有I/O信号
- C语言是顺序执行，而VerilogHDL的所有module均并发执行
- C语言与VerilogHDL语法相似性

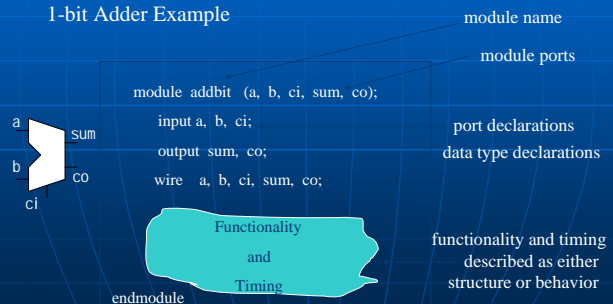
2.Module模块的基本结构

由关键词module和endmodule定义



举例1：1bit加法器

1-bit Adder Example



举例2 : D触发器

```
module dff_pos(data,clk,q);
  input data,clk;
  output q;
  reg q;
  always@(posedge clk)
    q=data;
endmodule
```



3.行为描述与结构描述

- VerilogHDL可以对电子系统在系统层或行为层进行行为性抽象描述，也可以针对具体的行为或功能进行硬件电路实现的结构性描述

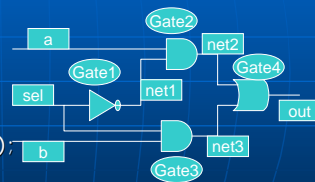
举例3 : 二选一电路MUX

- 行为描述

```
module mux_beh(out,a,d,sel);
  output out;
  input a,b,sel;
  assign out=(sel == 0)? a : b;
endmodule
```

- 结构描述

```
module mux_str(out,a,b,sel);
  output out;
  Input a,b,sel;
  not gate1(net1,sel);
  and gate2(net2,a,net1);
  and gate3(net3,b,sel);
  or gate4(out,net2,net3);
endmodule
```



not、and、or为VerilogHDL的基本门级元件，gatex为单元电路（门）名称，netx为电路内部连线名称。

4. 仿真与测试

- 通过测试模块对所设计的模块进行仿真与测试
- 测试模块须包含如下内容：
 - 对被测试模块调用
 - 给出测试矢量（激励信号）
 - 能对被测模块输出信号进行检测，并报告结果

举例4：对MUX模块的测试

```
module test_for_mux;
    reg a,b,sel;
    mux_str mux1(out,a,b,s);
    initial
    begin
        a=0;b=1;s=0;
        #10 a=1;
        #10 b=0;
        #10 s=1;
        #10 b=1;
        #10 a=0;
        #10 $finish
    end
    initial
        $monitor($time," a=%b b=%b s=%b
        out=%b",a,b,s,out);
endmodule
```

三、硬件设计实现

- VerilogHDL的设计实质是把电子系统的各功能模块进行Verilog描述，这种描述可以是行为描述，也可以是结构描述
- 通常，一个电子系统有多个不同的功能模块构成，但总有一个模块将所有模块连接起来，完成整个电子系统的协同工作，这个模块就是顶层模块
- 复杂电子系统通常由顶层向底层逐层展开设计，各功能模块的内部结构逐级得到深化和细化
- 在自顶向下的设计过程，均可以通过测试模块对每个层次的设计模块进行测试与评估

小结

- TopDown的设计过程
- VerilogHDL与C语言的异同
- VerilogHDL模块的主要构成
- 行为描述与结构描述
- 用VerilogHDL进行硬件设计的实现过程