

三 线上实验平台介绍

易学易用的开源虚拟仿真平台Logisim

- **采用原理图方式进行设计**
- 简单易学，易调试、零实验成本，**有效拓展实验时间**，是线下实验有力补充
- 支持子电路封装，易于构建复杂数字电路系统（CPU）
- 延续了数字逻辑课程中数字电路设计方法
 - 回避了HDL过于抽象、硬件设计程序化的问题
 - 有利于培养学生硬件设计思维
- 软件占用空间小、运行速度快、可扩展



Berkeley
UNIVERSITY OF CALIFORNIA



Cornell University



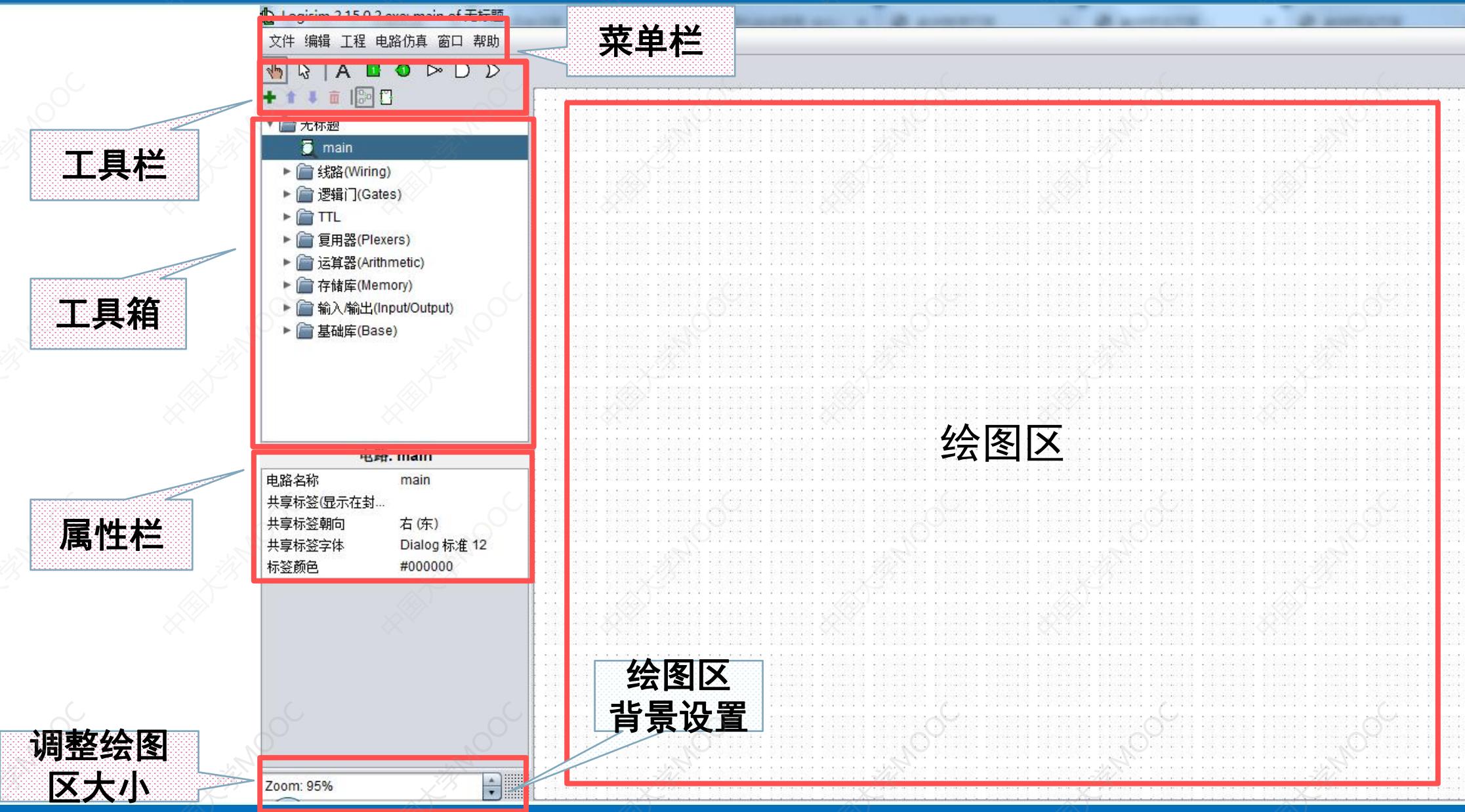
TEXAS A&M
UNIVERSITY

三 线上实验平台介绍

主要内容

1. logisim仿真软件主界面
2. logisim功能菜单
3. 用logisim设计电路步骤和流程
4. 一位全加器设计和仿真测试
5. 硬件重用与迭代设计思想

三 logisim仿真软件主界面



三 logisim丰富的资源库

- 线路
- 逻辑门
- TTL
- 复用器
- 运算器
- 存储库
- 输入输出
- 基础库



分线器(Splitter)

引脚(Pin)

探针(Probe)

隧道(Tunnel)

上/下拉电阻

时钟(Clock)

可编程发生器

常量(Constant)

电源(Power)

接地(Ground)

三极管

传输门(Transmission Gate)

位扩展器

非门(NOT Gate)

缓冲器(Buffer)

与门(AND Gate)

或门(OR Gate)

与非门(NAND Gate)

或非门(NOR Gate)

异或门(XOR Gate)

异或非(XNOR Gate)

奇校验(Odd Parity)

偶校验(Even Parity)

三态门(Controlled Buffer)

三态非门(Controlled Inverter)

三 logisim丰富的资源库

- 线路
- 逻辑门
- TTL
- 复用器
- 运算器
- 存储库
- 输入输出
- 基础库



运算器(Arithmetic)		输入/输出(Input/Output)	
⊕ 加法器(Adder)		□ 按钮(Button)	
⊖ 减法器(Subtractor)		▣ 开关(Switch)	
⊗ 乘法器(Multiplier)		▢ 拨码开关(Dip Switch)	
÷ 除法器(Divider)		▬ Slider	
⊹ 求补器(Negator)		● 摆杆(Joystick)	
⊸ 比较器(Comparator)		▤ 键盘(Keyboard)	
⊸ 移位器(Shifter)		● 蜂鸣器(Buzzer)	
⊕ 逐位加法器(Bit Adder)		● 发光二极管(LED)	
▢ 位查找器(Bit Finder)		● 彩色发光二极管(RGB LED)	
存储库(Memory)		数字示波器(Digital Oscilloscope)	
D 触发器(D Flip-Flop)		▢ 7段数码管(7-Segment Display)	
T 触发器(T Flip-Flop)		▢ 16进制数字显示(Hex Digit Display)	
JK JK触发器(J-K Flip-Flop)			
SR RS锁存器(S-R Flip-Flop)			
▢ 寄存器(Register)			
▢ 计数器(Counter)			
▢ 移位寄存器(Shift Register)			
● 随机数生成器(Random Generator)			
RAM 随机存储器(RAM)			
ROM 只读存储器(ROM)			
PLA 可编程逻辑门阵列(PLA ROM)			

三 logisim丰富的资源库

- 线路
- 逻辑门
- TTL
- 复用器
- 运算器
- 存储库
- 输入输出
- 基础库



TTL

- 7400: quad 2-input NAND gate
- 7402: quad 2-input NOR gate
- 7404: hex inverter
- 7408: quad 2-input AND gate
- 7432: quad 2-input OR gate
- 7447: BCD to 7-segment decoder
- 7485: 4-bit magnitude comparator
- 7486: quad 2-input XOR gate
- 74125: quad bus buffer, three-state outputs, negative enable
- 74165: 8-bit parallel-to-serial shift register
- 74283: 4-bit binary full adder
- 747266: quad 2-input XNOR gate

复用器(Plexers)

- ▷ 多路选择器(Multiplexer)
- ◁ 解复用器(Demultiplexer)
- ◁ 译码器(Decoder)
- 7段数码管译码器(7-Segment Display Decoder)
- Pri 优先编码器(Priority Encoder)
- ▷ 位选择器(Bit Selector)

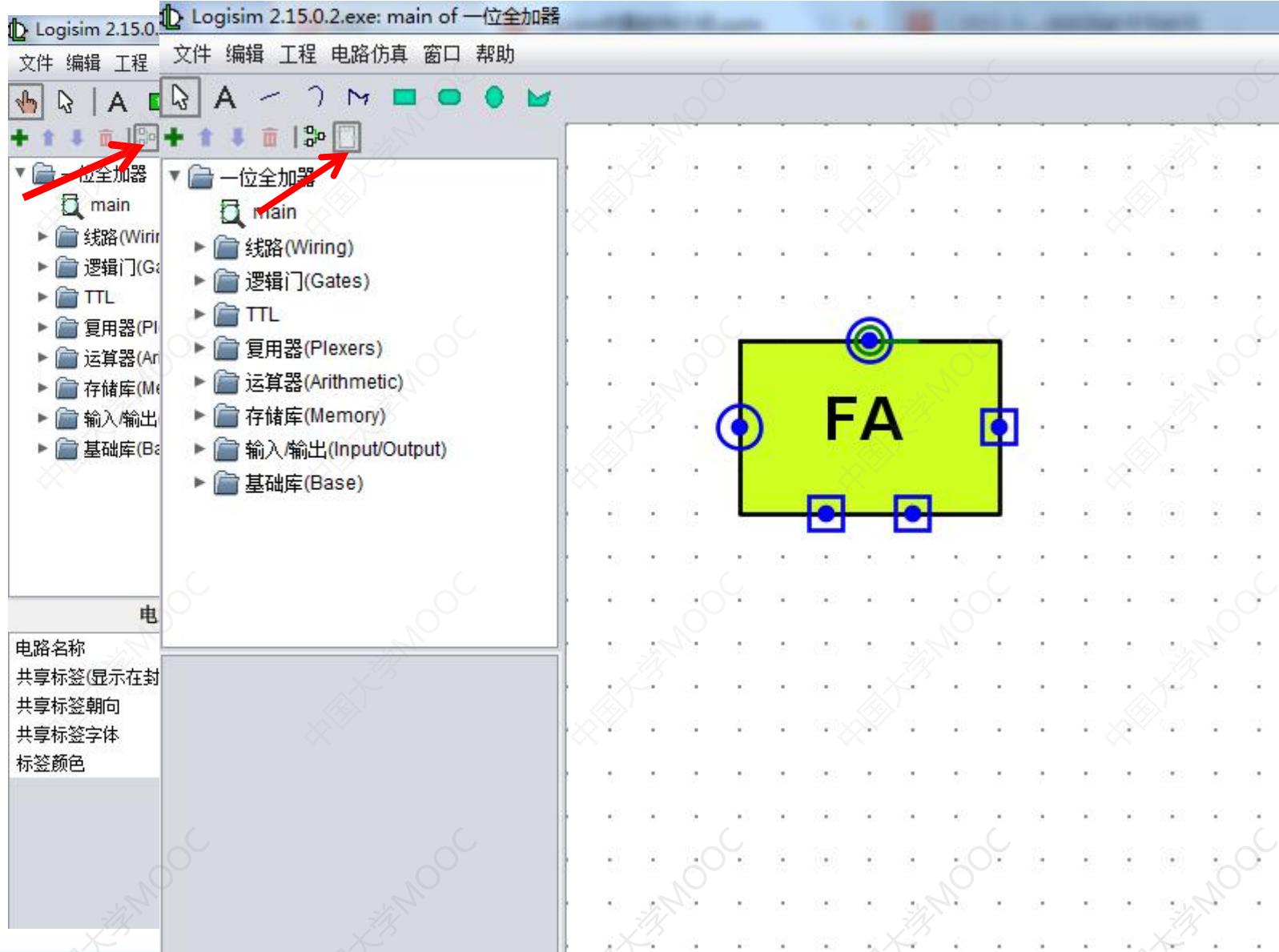
基础库(Base)

- 🕒 钳工具(Poke Tool)
- 🖱 编辑工具
- 🖱 选择工具
- 🖌 连线工具
- ✍ 文本工具
- 📋 菜单工具
- 🏷 标签(Label)

三 编辑模式和设计方法

■ 绘图区有两种设计模式：

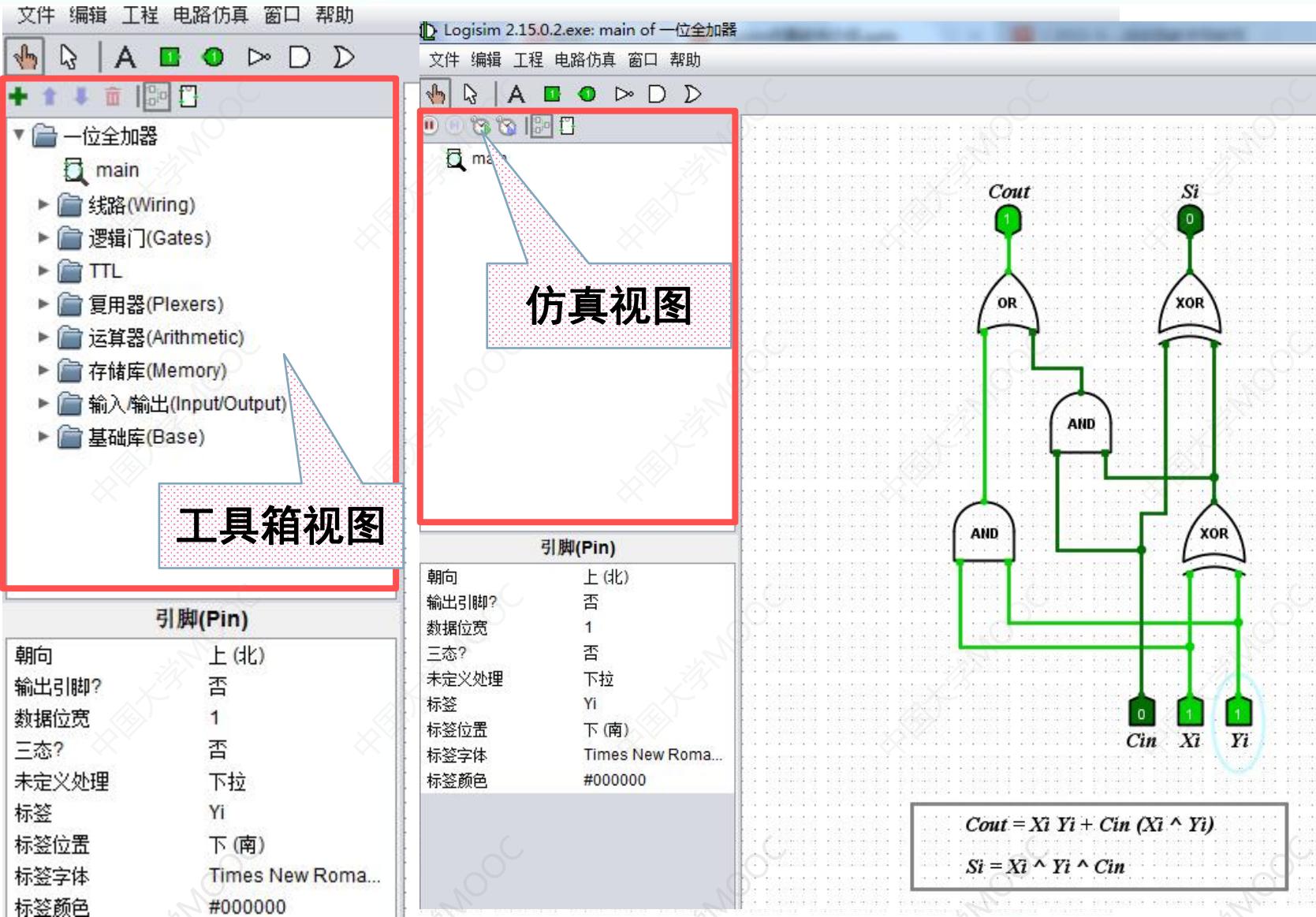
- 电路编辑模式
- 封装编辑模式



三 编辑模式和设计方法

■ 两种视图

- 工具箱视图（默认）
- 仿真视图、
点击“工程”-
“查看仿真视图”



三 编辑模式和设计方法

■两种绘制方法：

- 原理图直接绘制 电路编辑模式 工具箱视图
- 分析组合逻辑电路自动绘制

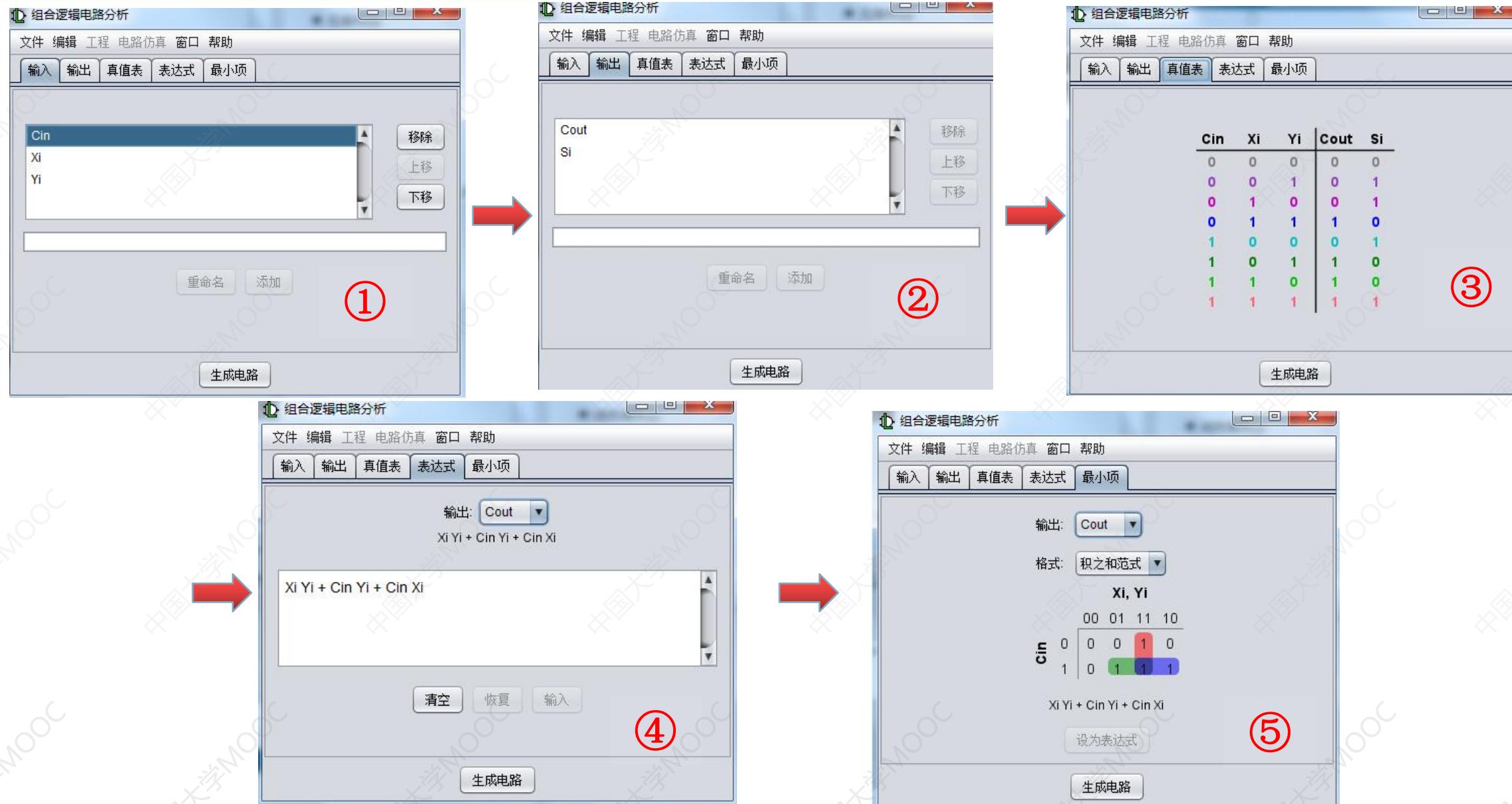
点击菜单”工程”→ “分析组合逻辑电路自动绘制”

具体步骤：

- (1) 设置输入变量
- (2) 设置输出变量
- (3) 填写真值表
- (4) 生成表达式
- (5) 简化表达式
- (6) 生成电路

其中，第5步可以跳过直接生成电路。

三 编辑模式和设计方法



三 编辑模式和设计方法

仿真

- 两种仿真模式
 - 工具箱视图（默认）

■ 戳工具 **ctrl+1**

设置输入或各组件值

ctrl+r 清零（电路复位）

ctrl+i 单步信号传递

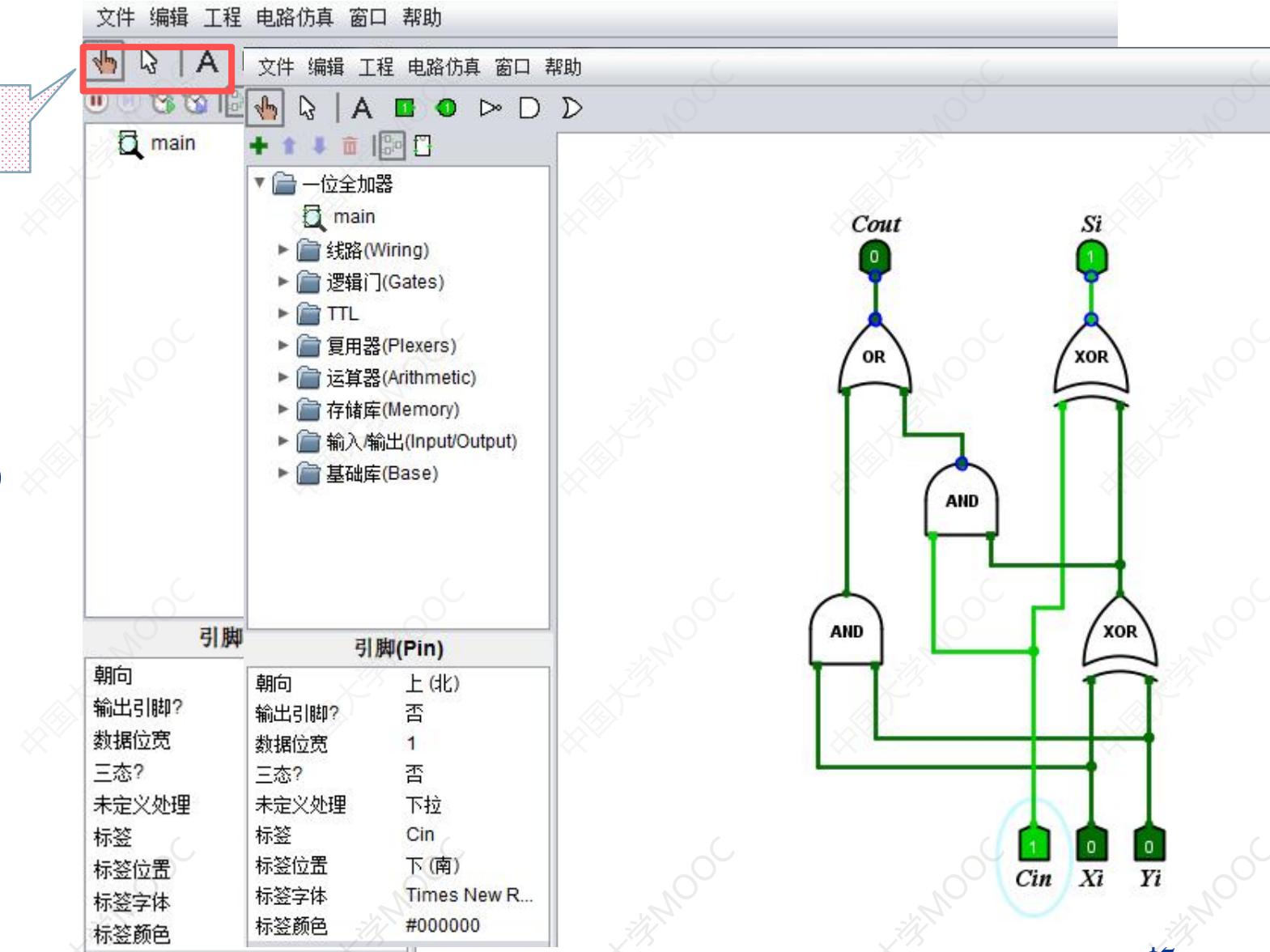
ctrl+t 单步运行

ctrl+k 连续/停止运行

- 仿真视图

点击菜单“工程”子菜单
“查看仿真视图”

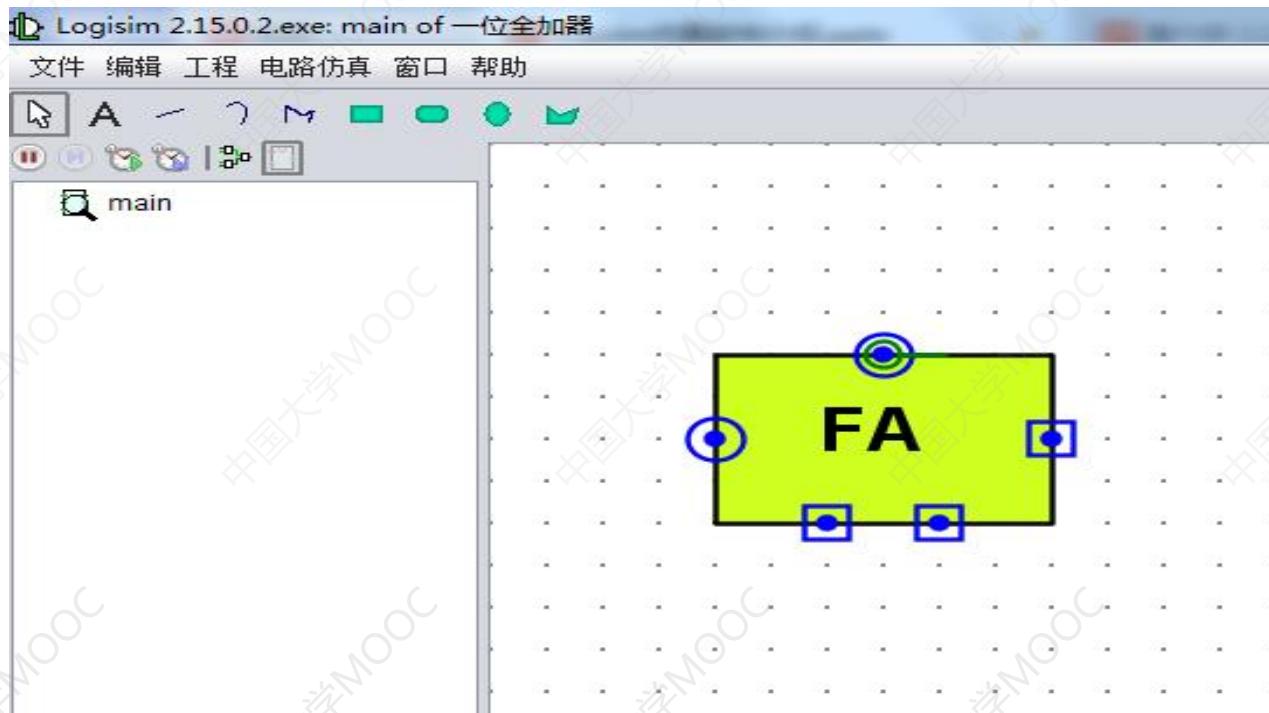
戳工具



三 封装测试

封装

- 仿真运行通过后，若要实现组件的重用，就必须对电路进行封装、测试。
- 点击封装编辑模式，就可以看到默认封装图，此时可以根据需要调整芯片外观、大小，引脚位置、名称，添加芯片名称等。



三 硬件重用与迭代设计

■ 硬件迭代设计思想

- 在设计复杂的硬件电路系统中，为方便设计和组件重用，通常采取自下而上顶层设计的思想，根据需求确定整体结构后，分模块实现，再连成整体进行联调。
- 若模块功能较为复杂、涉及引脚较多，可以将复杂的电路拆分成更简单、更小的电路模块设计实现后，再链接和组装成更大的电路模块，自下而上、层层封装、作为组件直接连接使用，此为组件重用和迭代设计。
- 此时各个模块间输入输出引脚定义和封装格式必须统一，即采用相同的接口，方便调试和迭代设计、升级。

■ 运算器设计为例

