



中山大學 软件工程学院
SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

计算机组成原理实验

授课老师：吴炜滨

大纲



- Logisim 基本功能
- Logisim中的延迟和险象
- Logisim中的震荡现象
- Logisim常用组件库



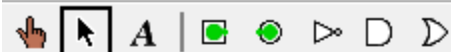
➤ Logisim 基本功能

- 分析电路

■ 画出如图电路

Logisim: 电路分析 of logisim基本功能

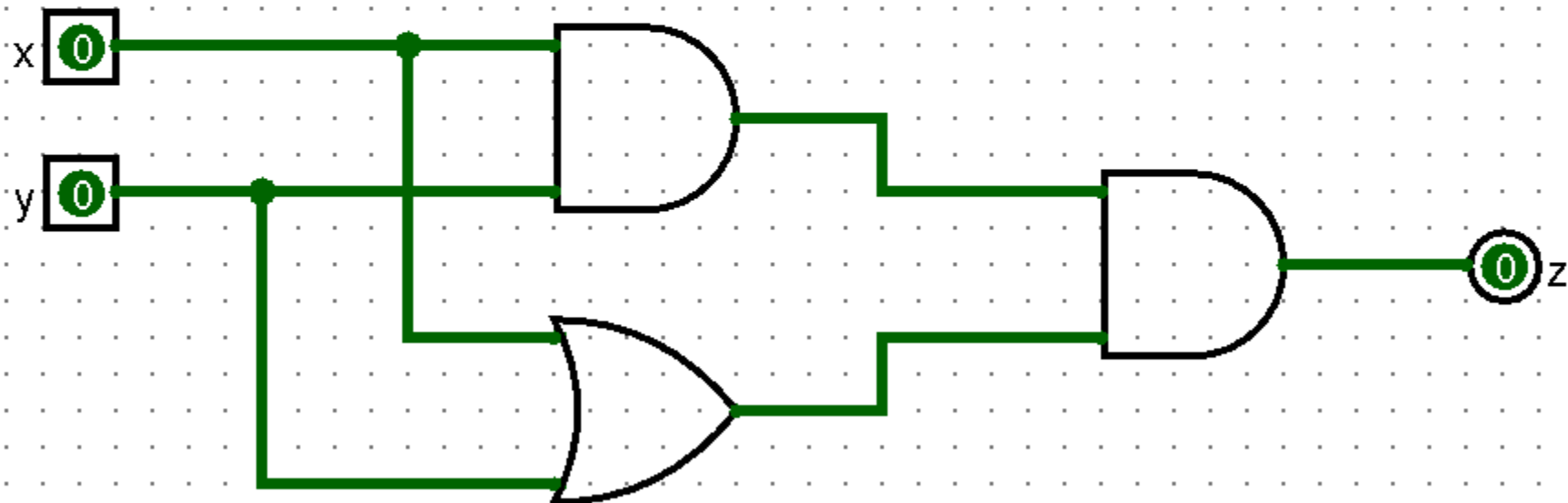
文件 编辑 项目 模拟 窗口 帮助



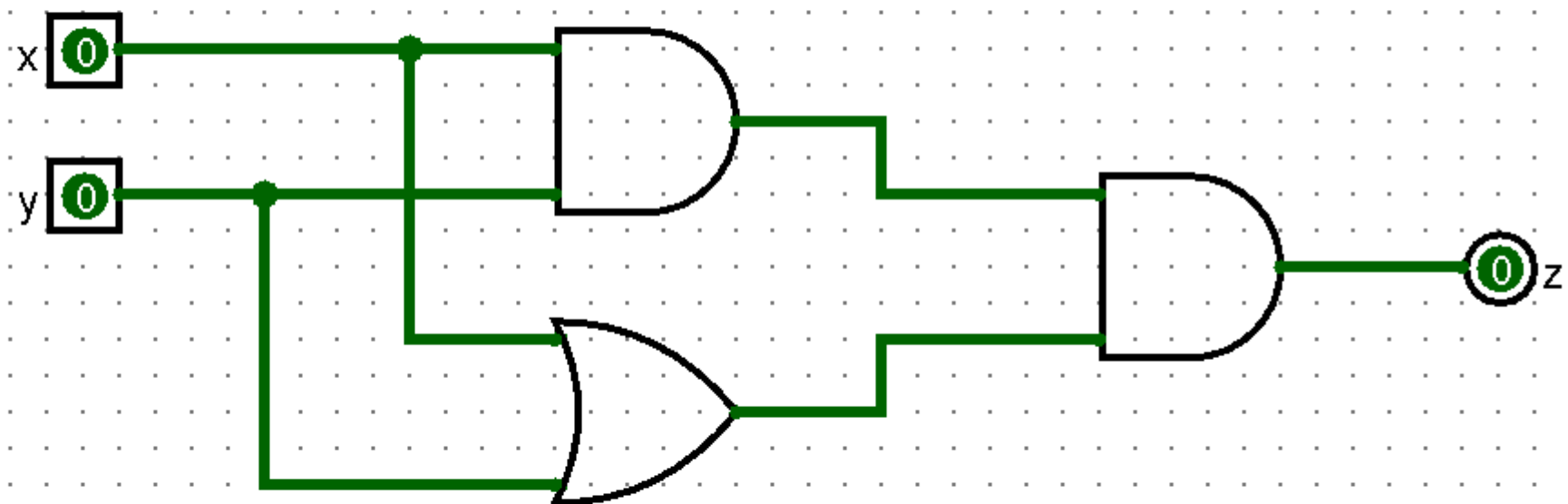
- logisim基本功能*
- 电路分析
- 线路(Wiring)
- 逻辑门(Gates)
- 复用器(Multiplexers)
- 运算器(Arithmetic)
- 存储(Memory)
- 输入/输出(Input/Output)
- 基本(Base)

电路: 电路分析

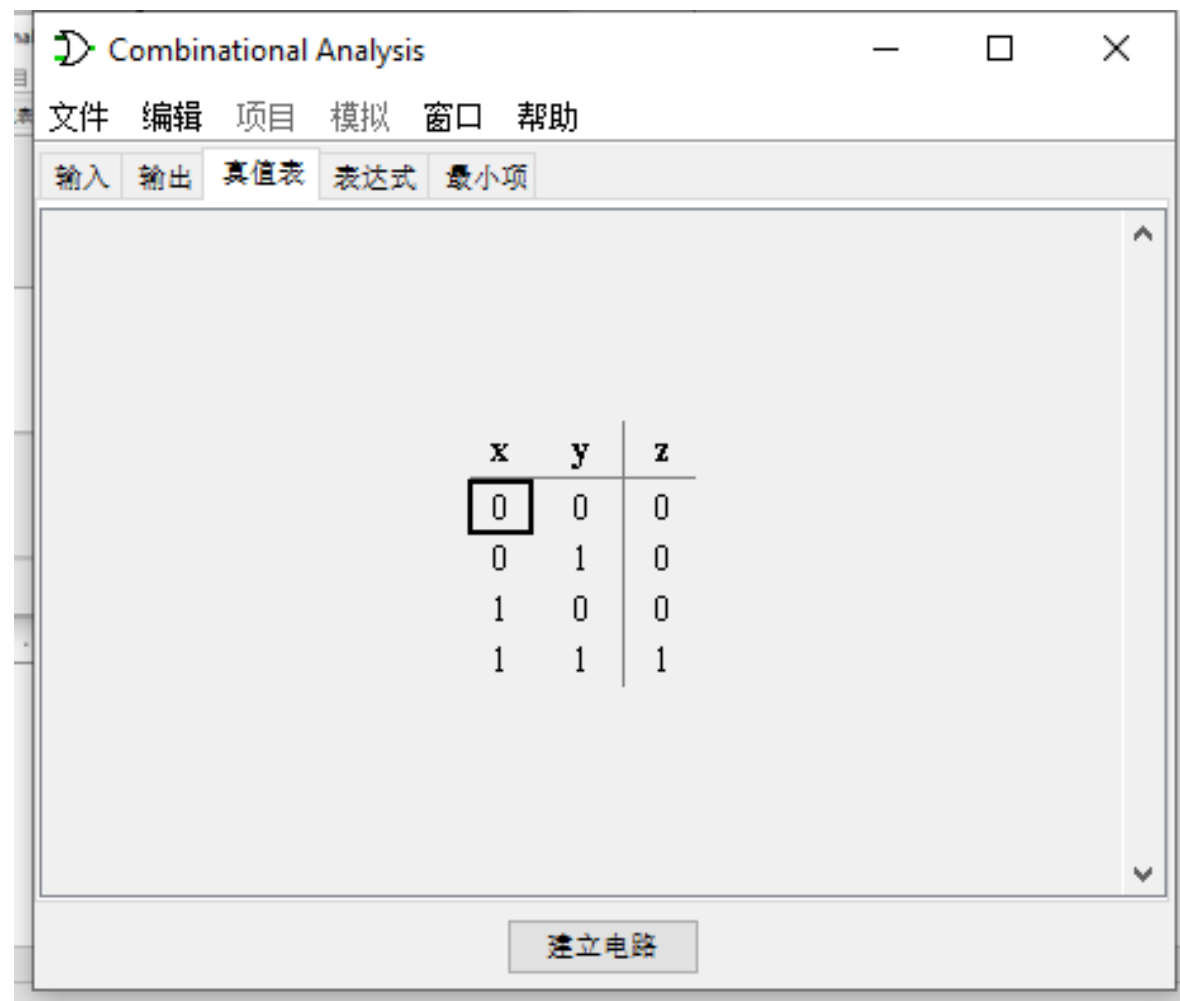
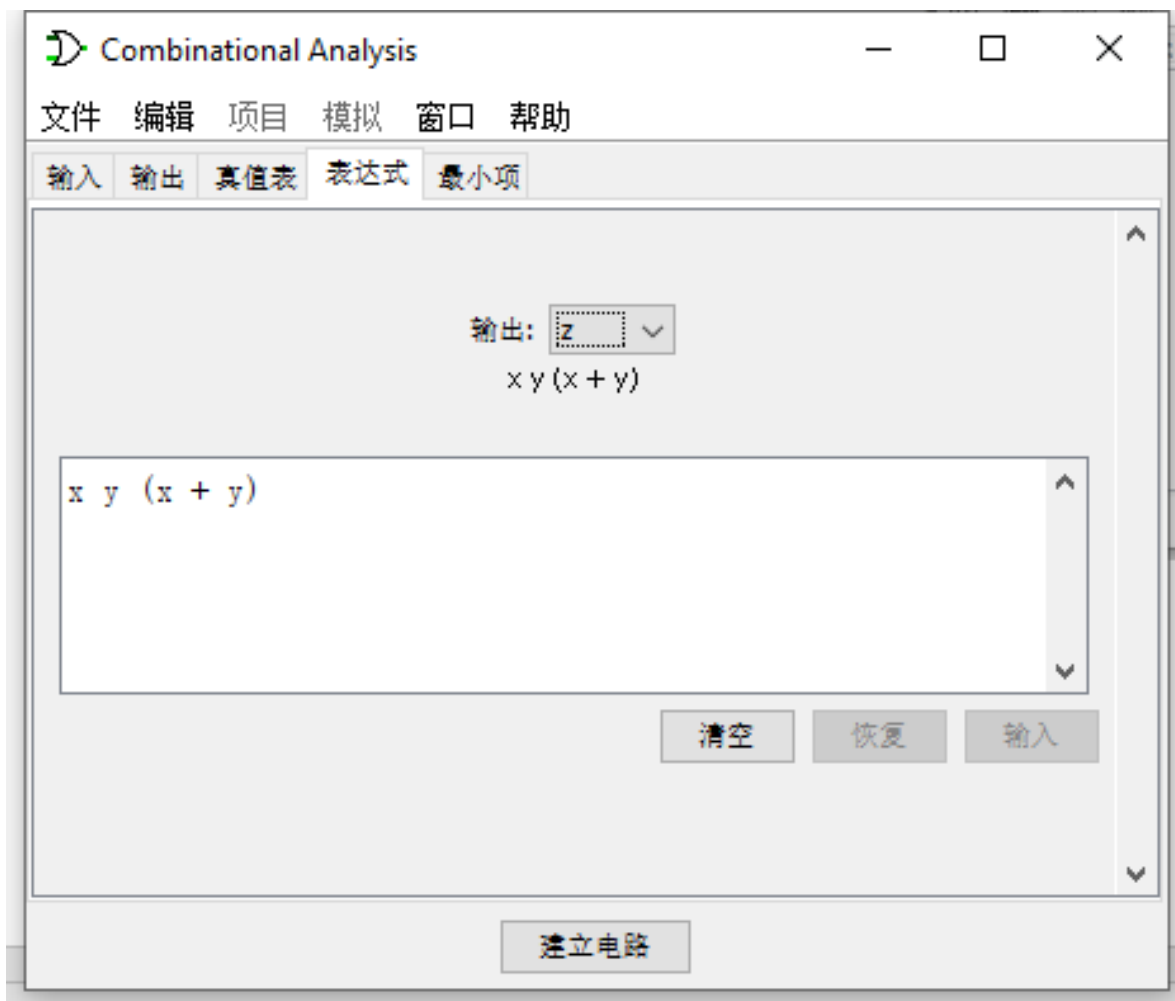
电路名称	电路分析
共享的标签	
共享的标签朝向	东
共享的标签字体	SansSerif 标准 12



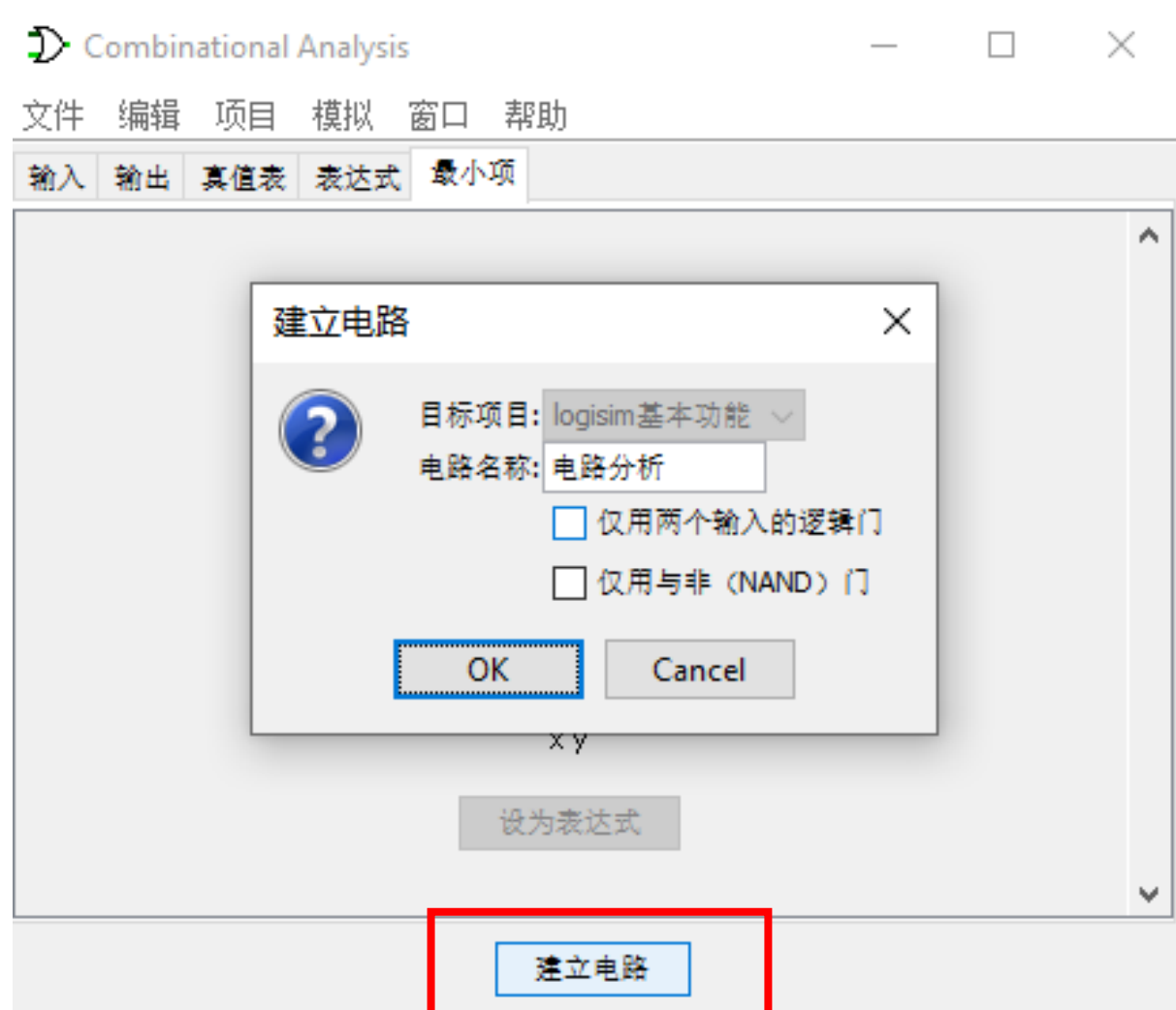
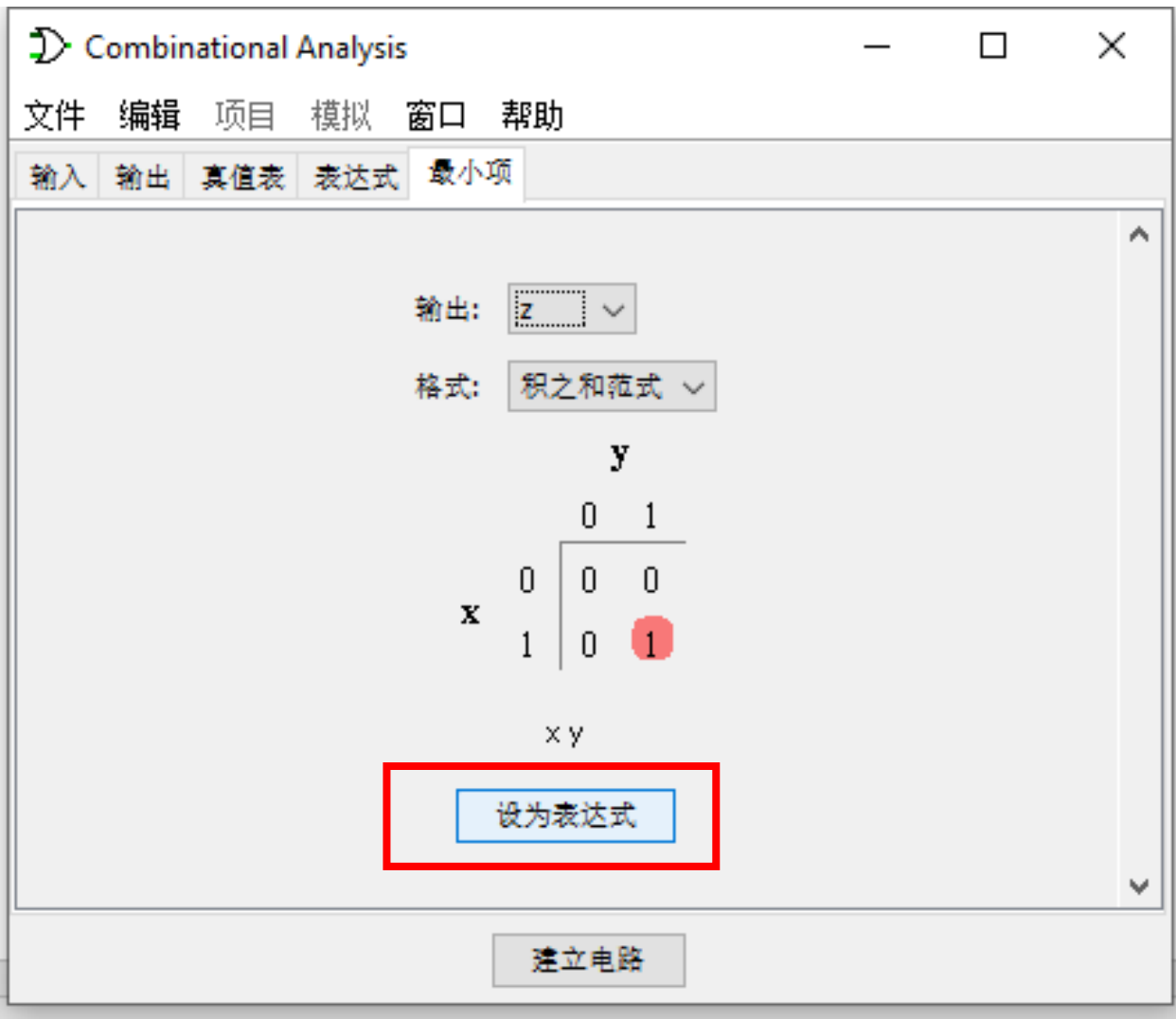
添加电路...
 加载库
 卸载库...
 向上移动电路
 向下移动电路
 设置为主电路
 移除电路
 恢复至默认外观
 查看工具箱
 查看模拟视图
 编辑电路布局
 编辑电路外观
 分析电路
 查看电路统计
 选项...

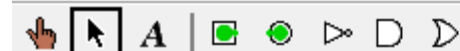


■ 电路表达式



■ 自动化简电路

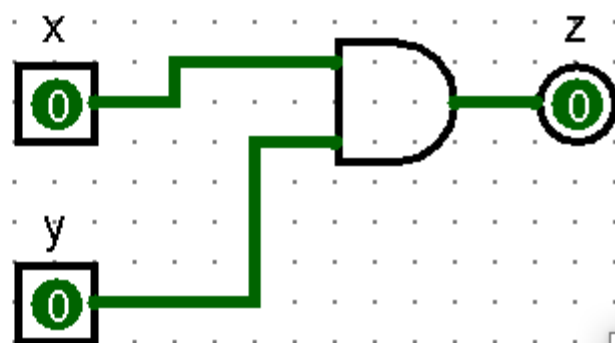




- logisim基本功能*
- 电路分析
- 线路(Wiring)
- 逻辑门(Gates)
- 复用器(Multiplexers)
- 运算器(Arithmetic)
- 存储(Memory)
- 输入/输出(Input/Output)
- 基本(Base)

电路: 电路分析

电路名称	电路分析
共享的标签	
共享的标签朝向	东
共享的标签字体	SansSerif 标准 12



Combinational Analysis

文件 编辑 项目 模拟 窗口 帮助

输入 输出 真值表 表达式 最小项

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

建立电路

■ 实时仿真开关

Logisim: 电路分析 of logisim基本功能

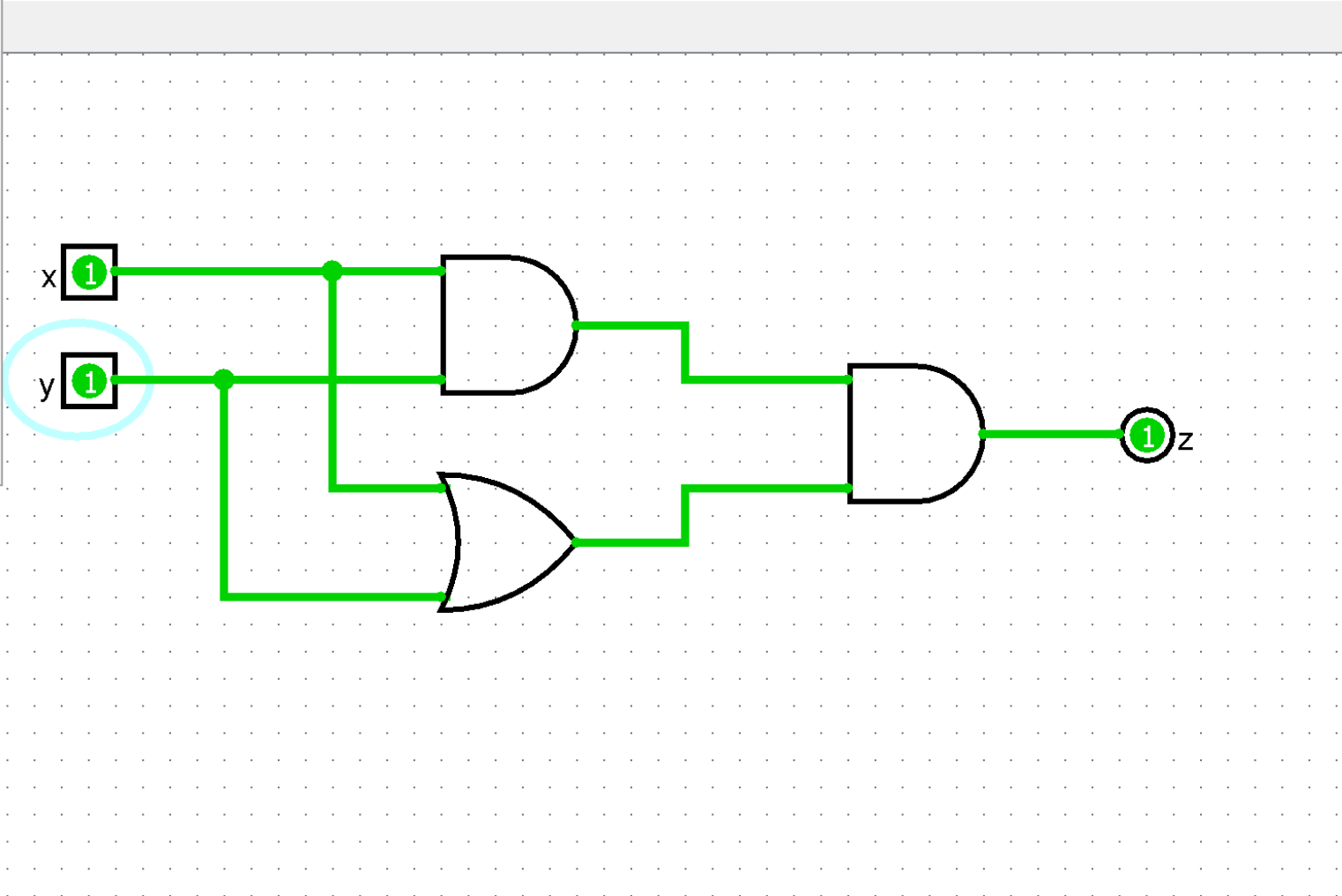
文件 编辑 项目 模拟 窗口 帮助



- logisim基本功能
 - 电路分析
 - 线路(Wiring)
 - 逻辑门(Gates)
 - 复用器(Plexers)
 - 运算器(Arithmetic)
 - 存储(Memory)
 - 输入/输出(Input/Output)
 - 基本(Base)

- ✓ 启用信号模拟 Ctrl+E
- 重置模拟器 Ctrl+R
- 信号传递一步 Ctrl+I
- 退出到 >
- 进入到 >
- 时钟前进一步 Ctrl+T
- 启用时钟模拟 Ctrl+K
- 时钟频率 >
- 记录器...

引脚(Pin)	
朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	y
标签位置	西
标签字体	SansSerif 标准 12



■ 实时仿真开关

Logisim: 电路分析 of logisim基本功能

文件 编辑 项目 模拟 窗口 帮助

logisim基本功能

电路分析

线路(Wiring)

逻辑门(Gates)

复用器(Plexers)

运算器(Arithmetic)

存储(Memory)

输入/输出(Input/Output)

基本(Base)

启用信号模拟 Ctrl+E

重置模拟器 Ctrl+R

信号传递一步 Ctrl+I

退出到 >

进入到 >

时钟前进一步 Ctrl+T

启用时钟模拟 Ctrl+K

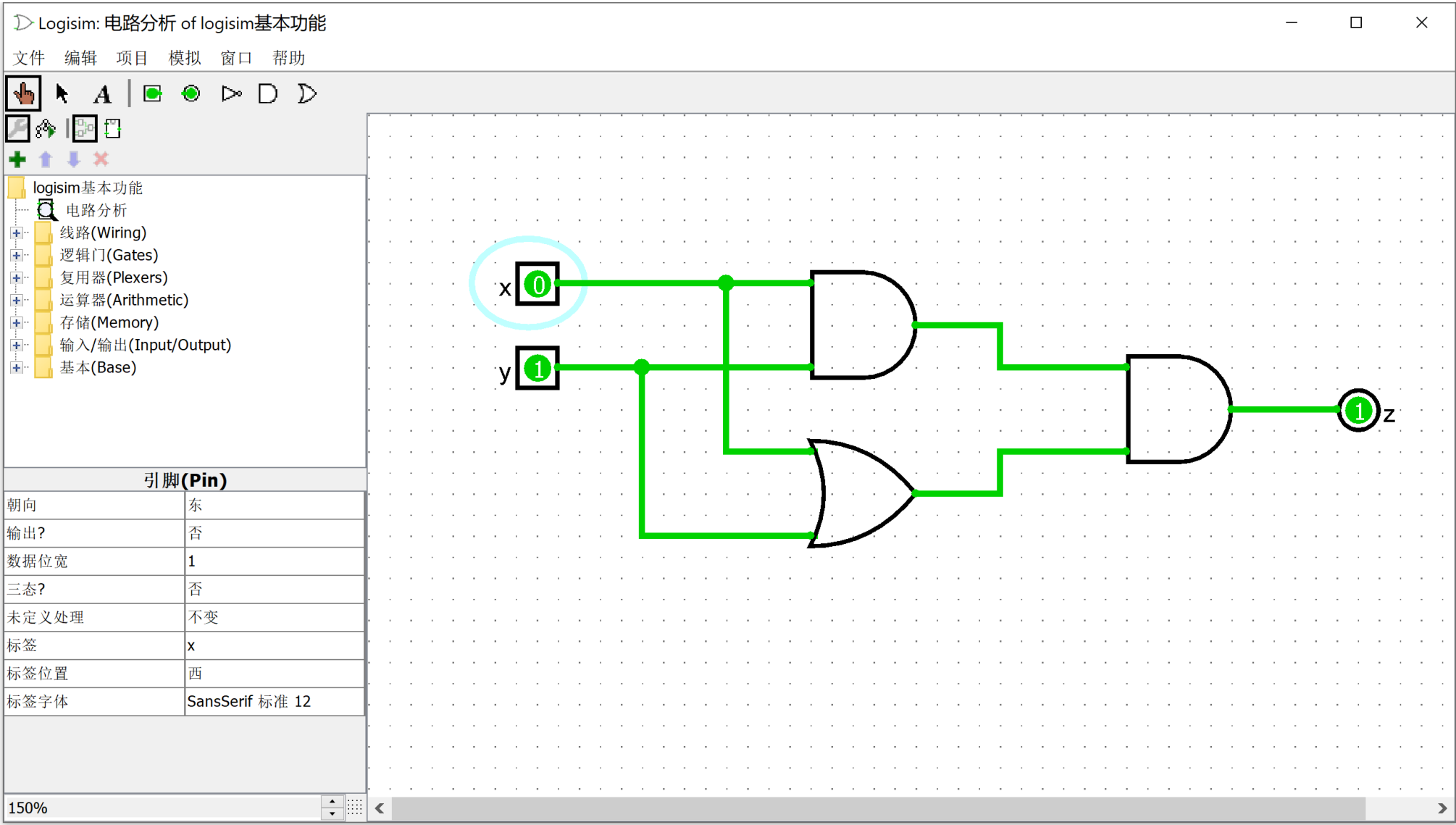
时钟频率 >

记录器...

引脚(Pin)	
朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	y
标签位置	西
标签字体	SansSerif 标准 12

150%

■ 实时仿真开关



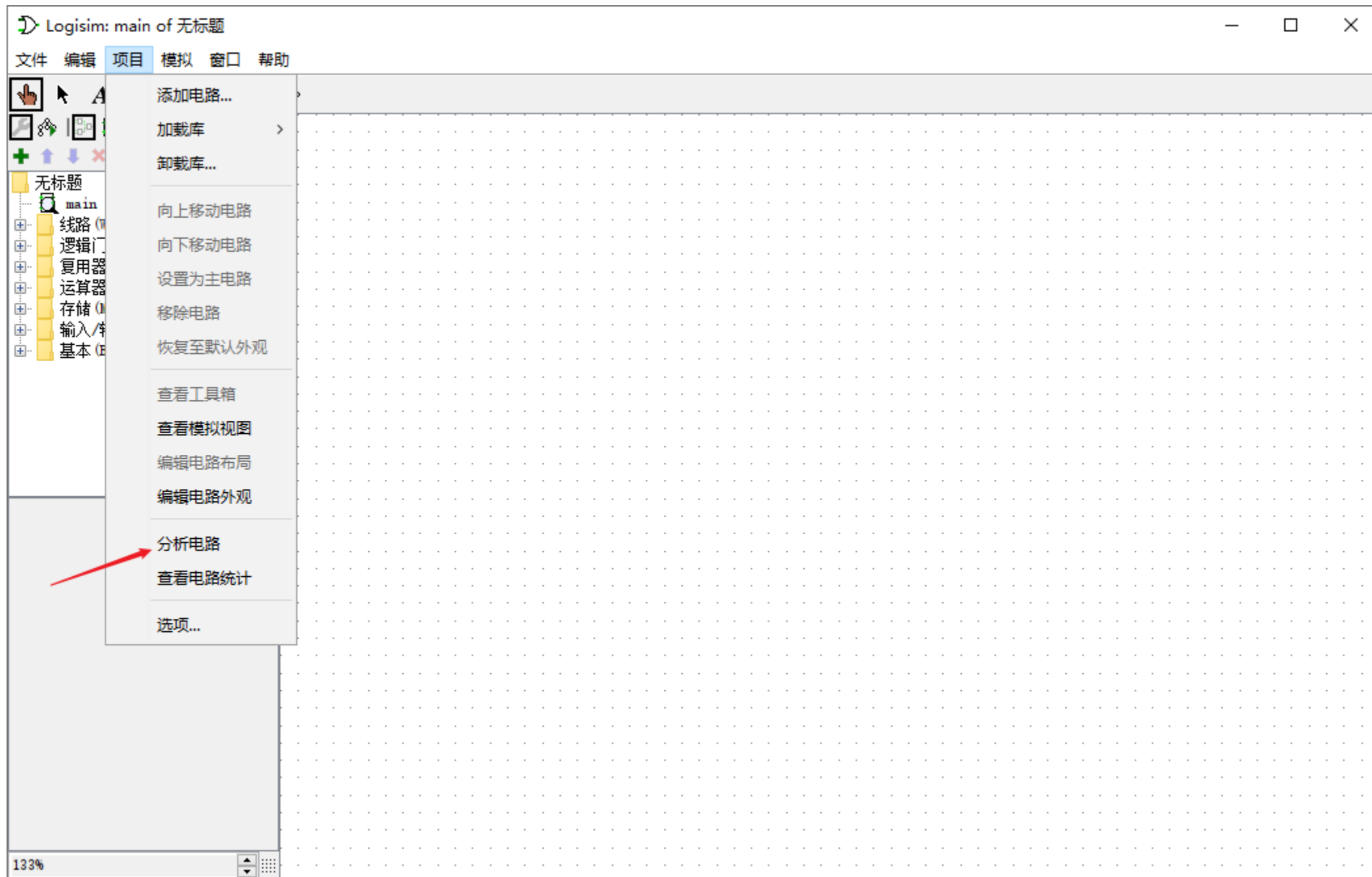


- Logisim 基本功能
 - 自动生成电路

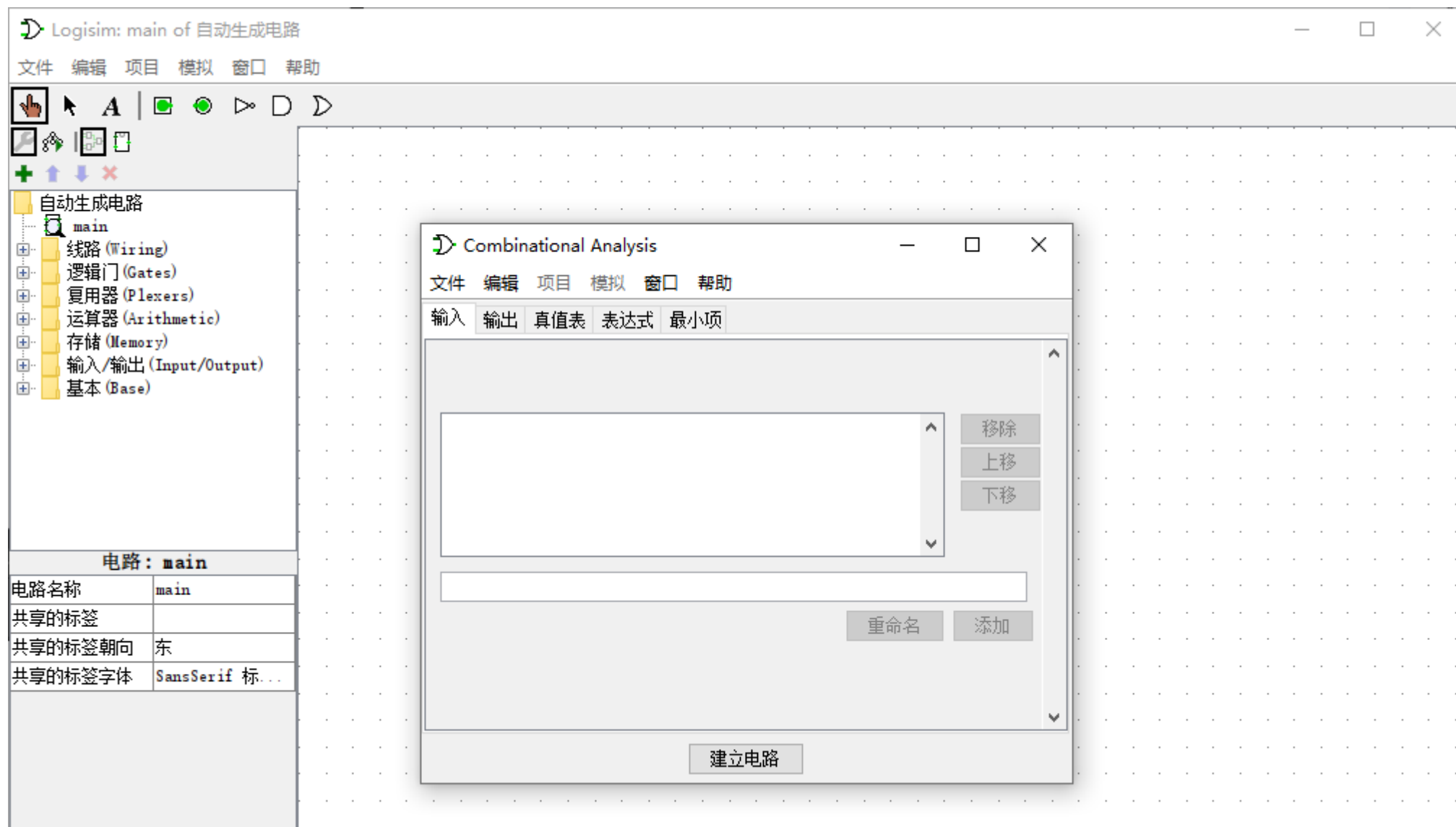
自动生成电路



- 利用真值表及表达式自动生成电路，但是只能是组合逻辑
- 输入引脚只能是一位
- 输入引脚最多是8个
- 输出引脚最多是12个

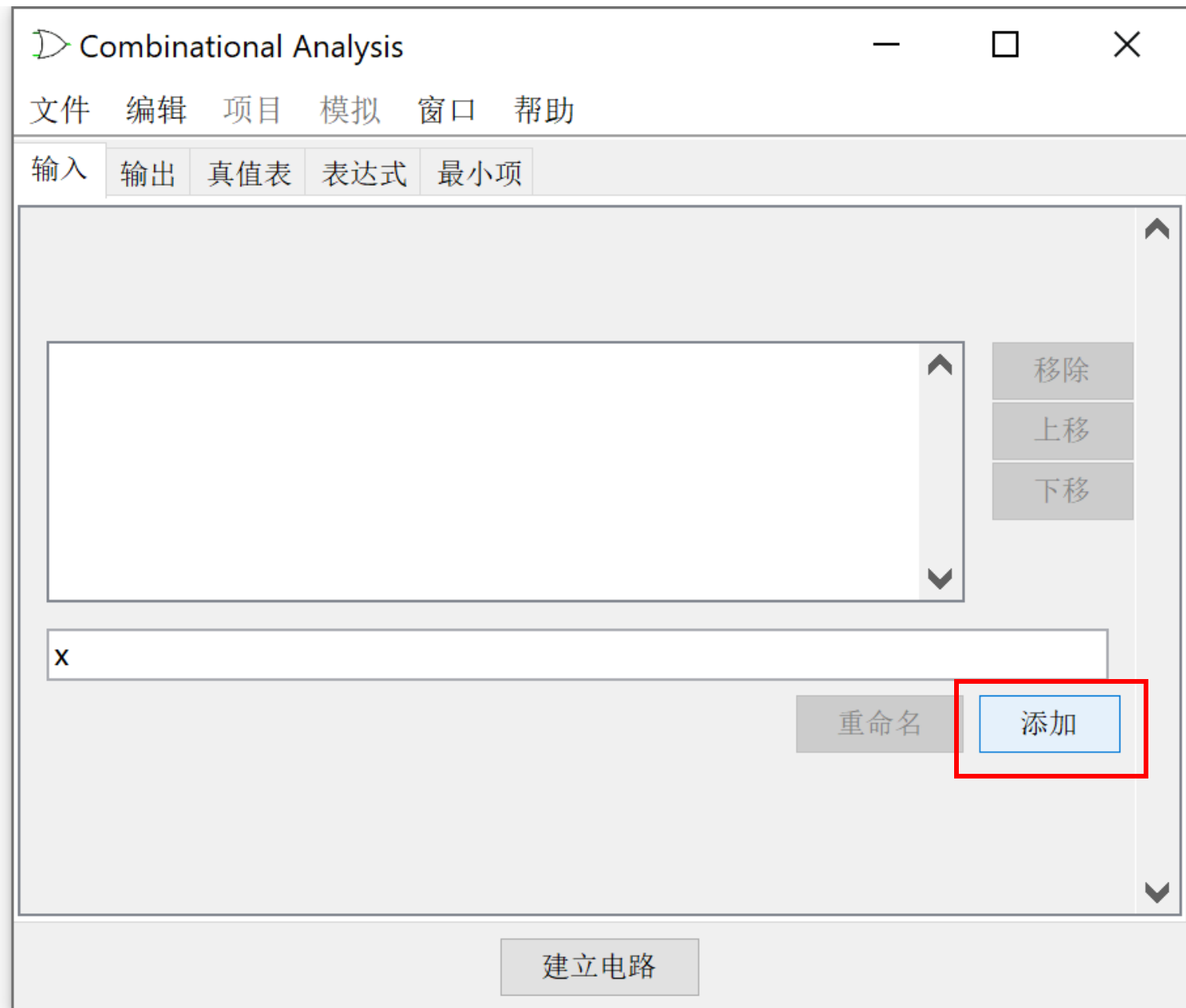


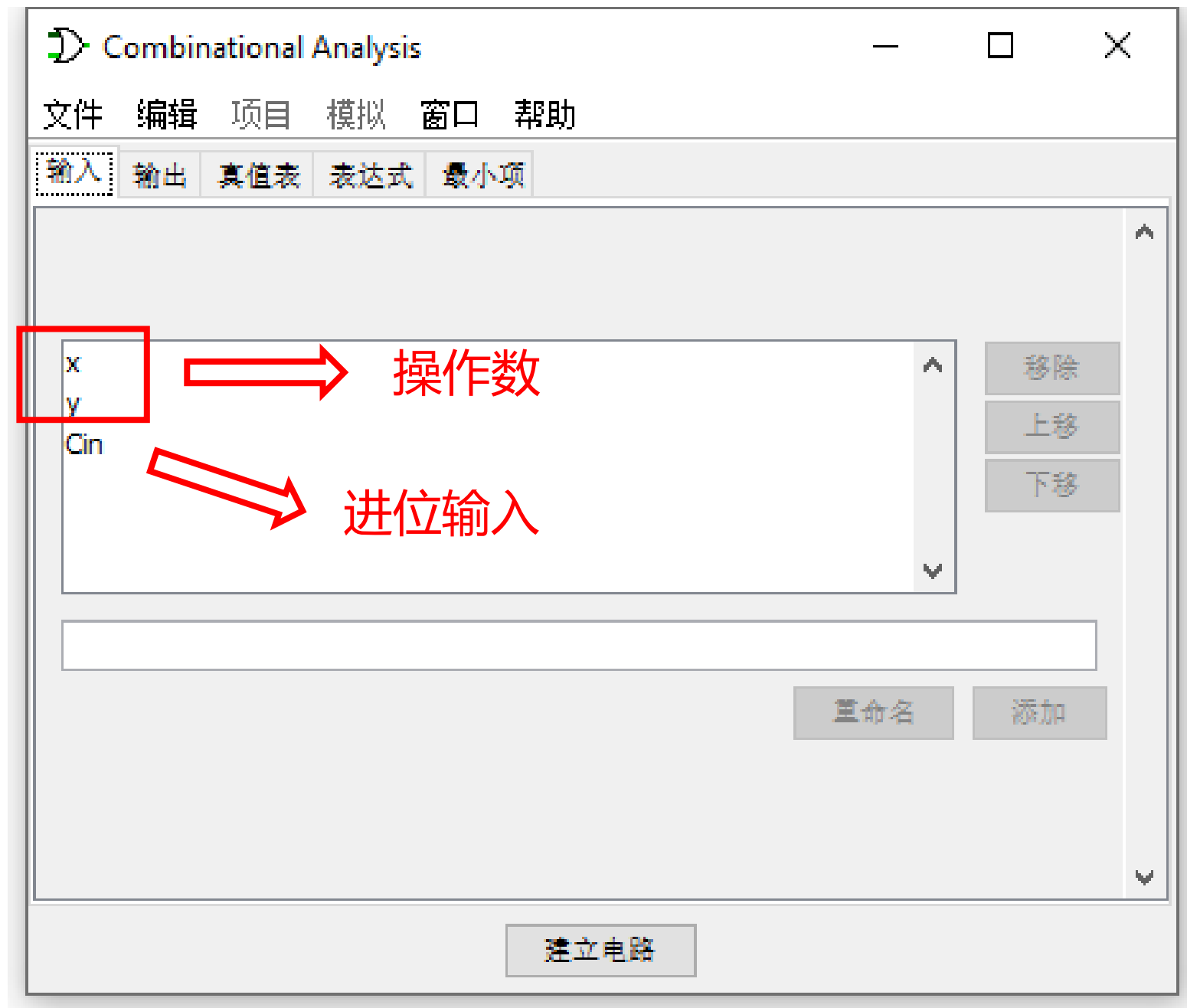
一位全加器

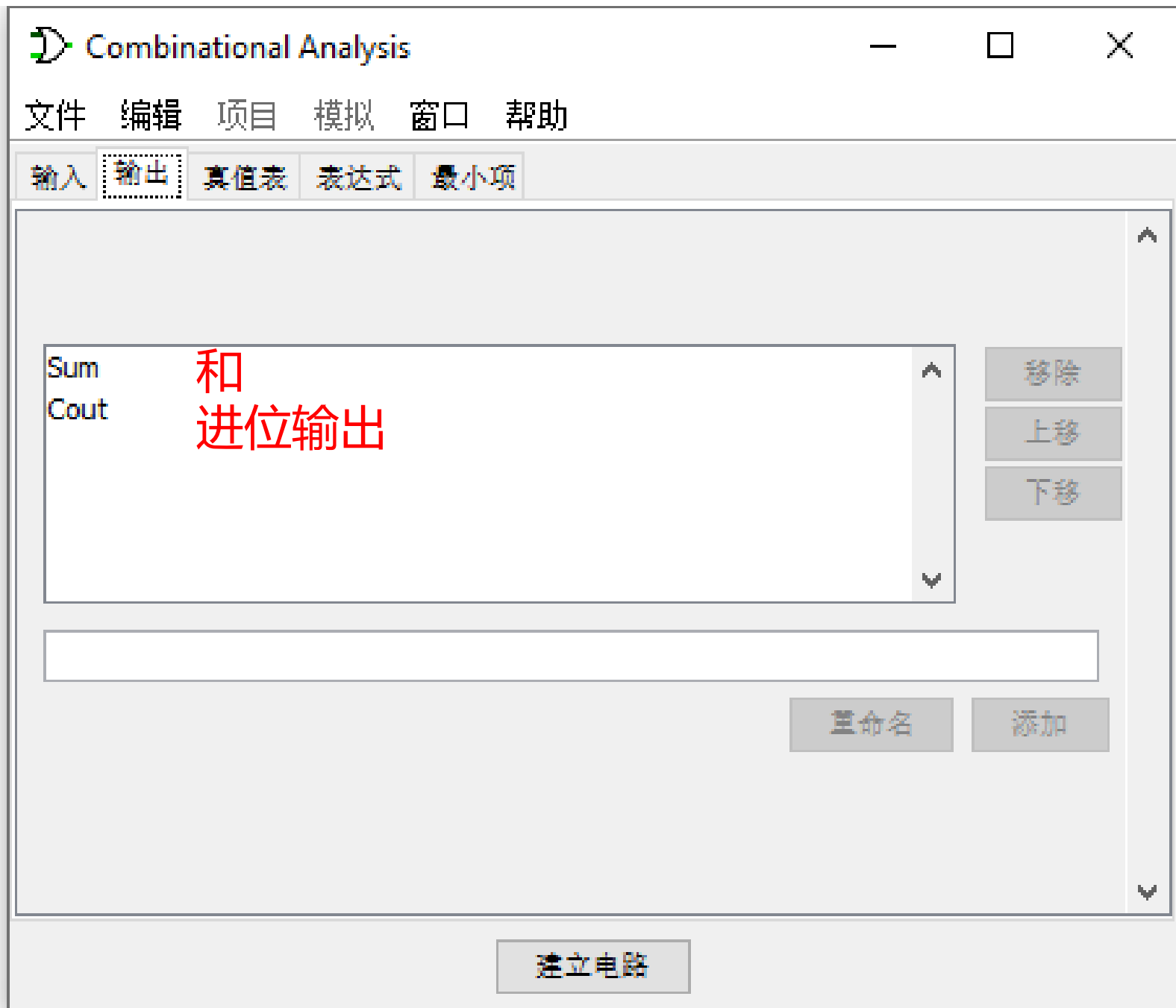


■ 一位全加器

- 添加输入/输出变量

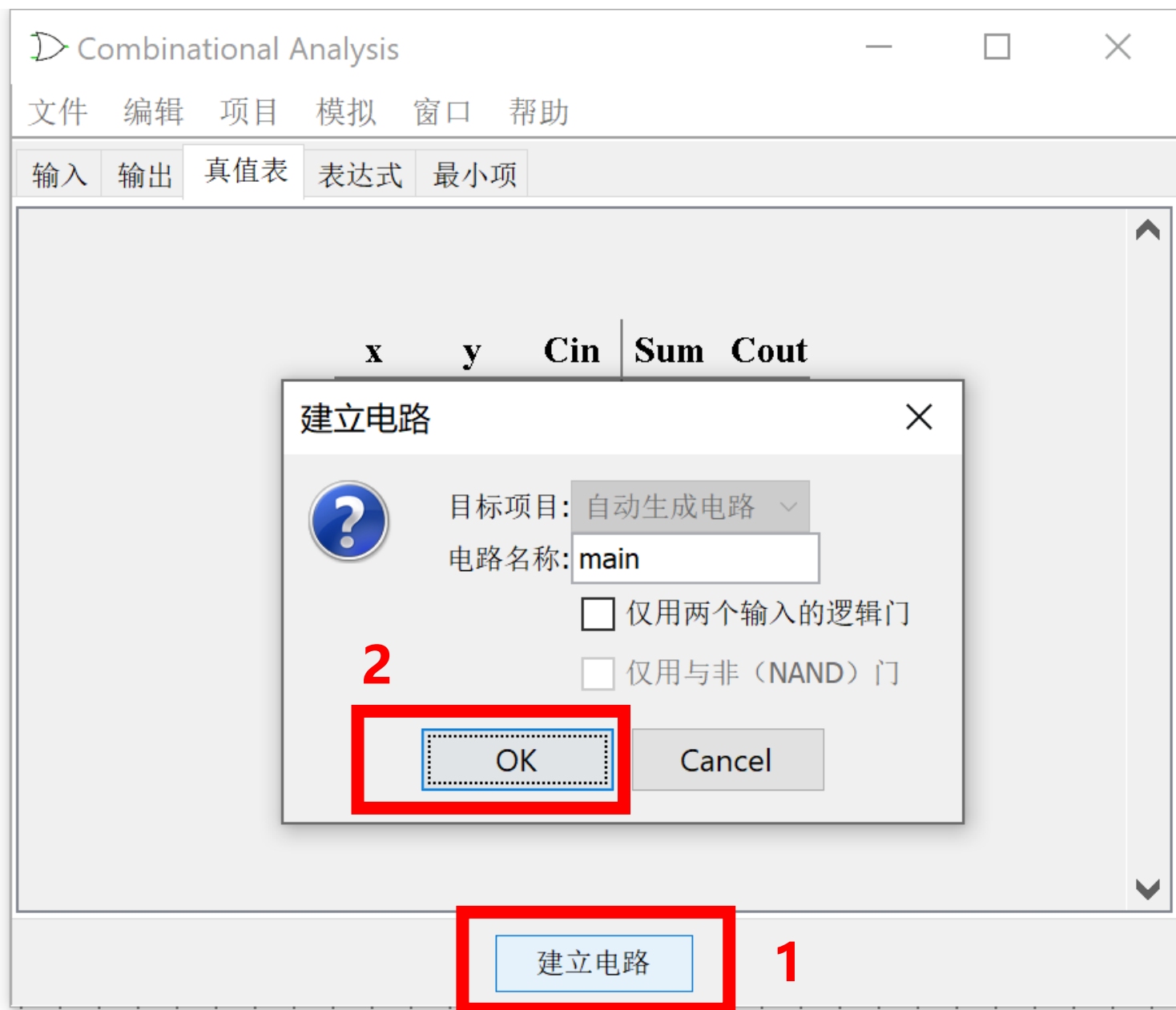


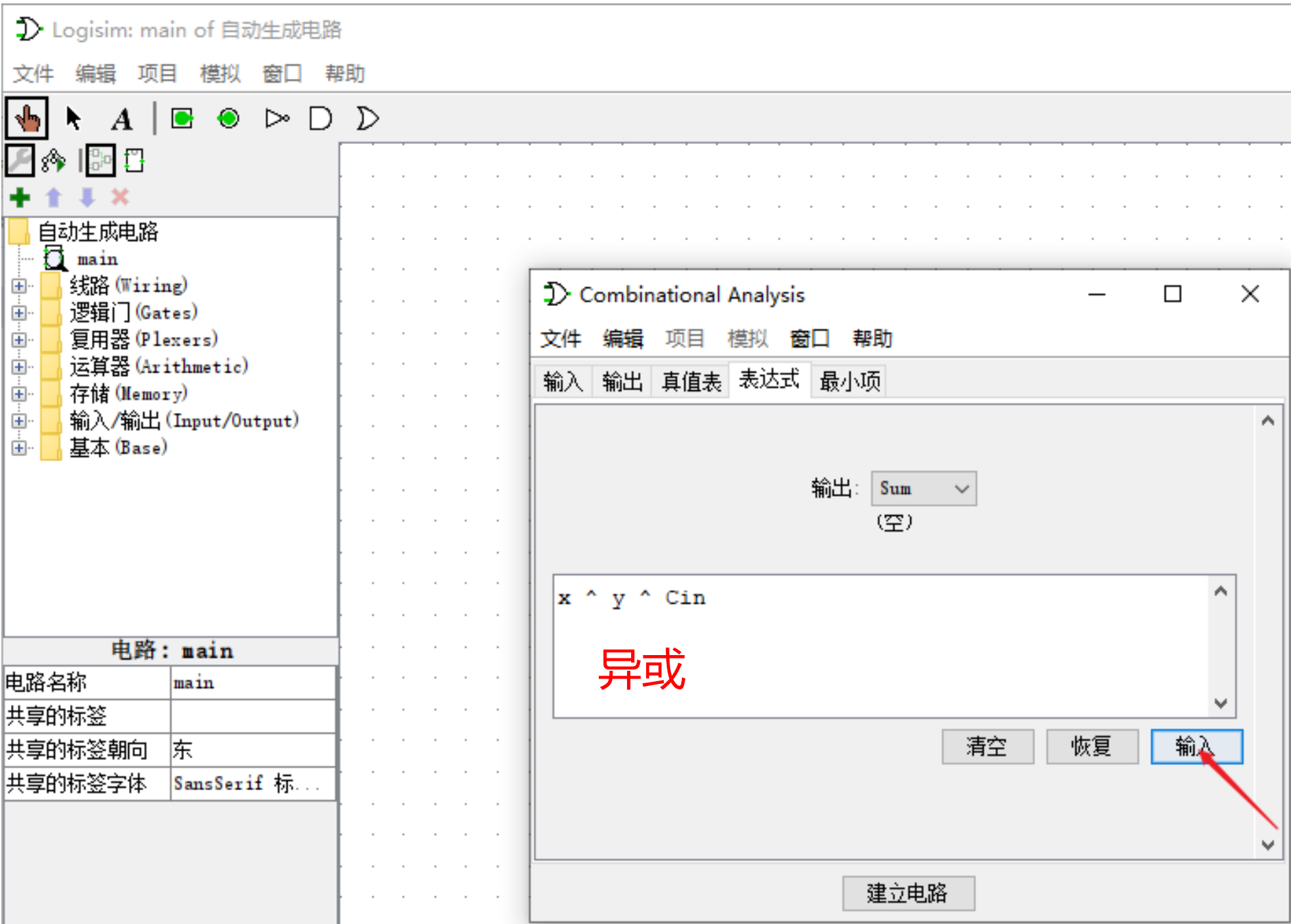






可以定义
真值表

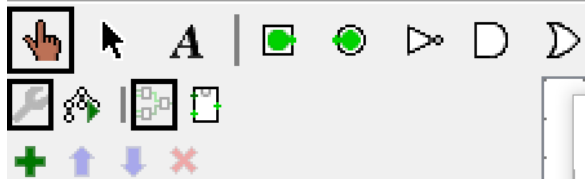




利用输入
表达式定
义电路

Logisim: main of 自动生成电路

文件 编辑 项目 模拟 窗口 帮助



- 自动生成电路*
- main
- 线路(Wiring)
- 逻辑门(Gates)
- 复用器(Plexers)
- 运算器(Arithmetic)
- 存储(Memory)
- 输入/输出(Input/Output)
- 基本(Base)

Combinational Analysis

文件 编辑 项目 模拟 窗口 帮助

输入 输出 真值表 表达式 最小项

输出: Sum
 $x \wedge y \wedge \text{Cin}$

$x \wedge y \wedge \text{Cin}$

清空 恢复 输入

建立电路



- 自动生成电路*
- main
- 线路(Wiring)
- 逻辑门(Gates)
- 复用器(Plexers)
- 运算器(Arithmetic)
- 存储(Memory)
- 输入/输出(Input/Output)
- 基本(Base)

Combinational Analysis

输入 输出 真值表 表达式 最小项

输出: Cout
 $x y + x \text{ Cin} + y \text{ Cin}$

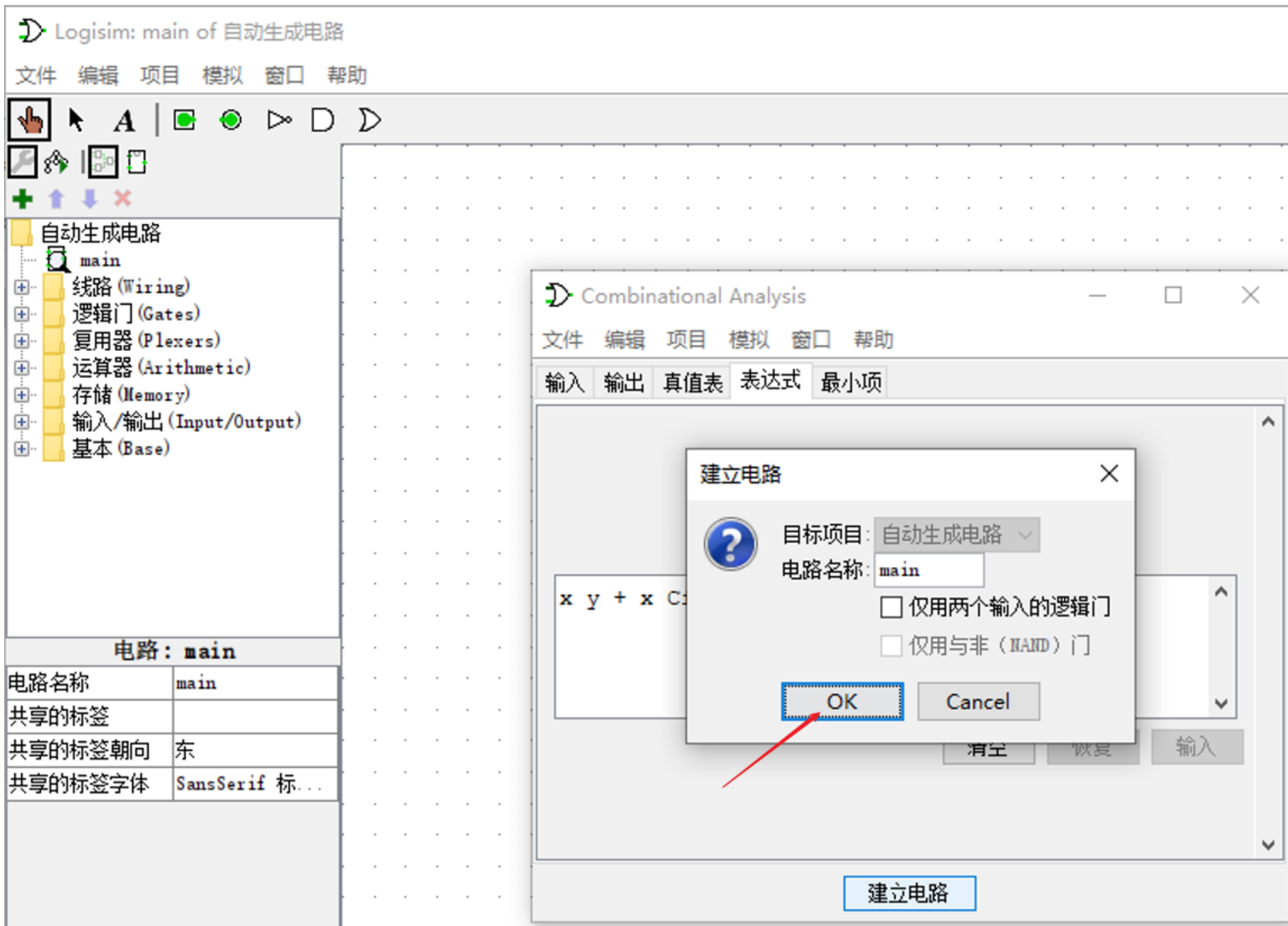
$x y + x \text{ Cin} + y \text{ Cin}$

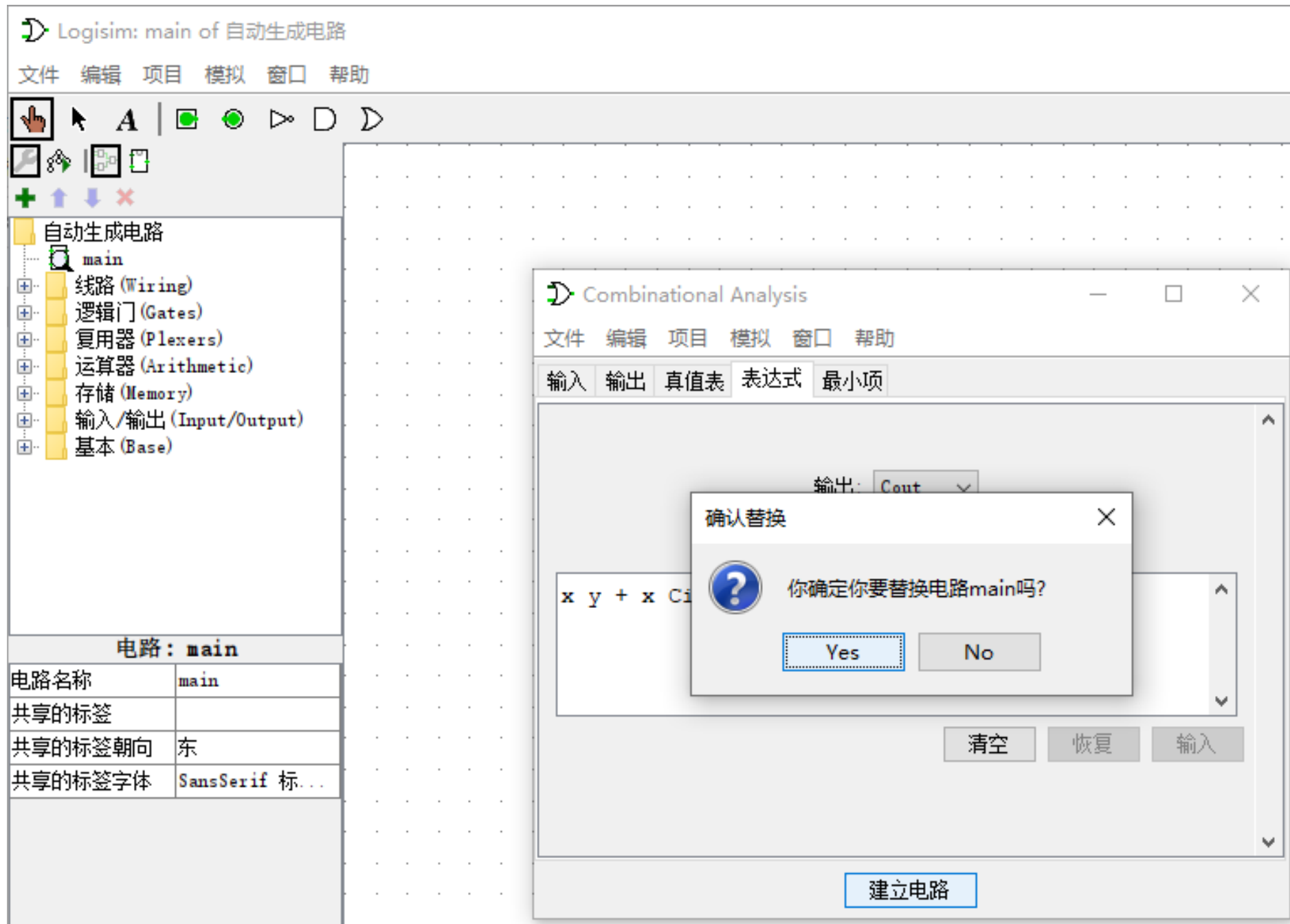
清空

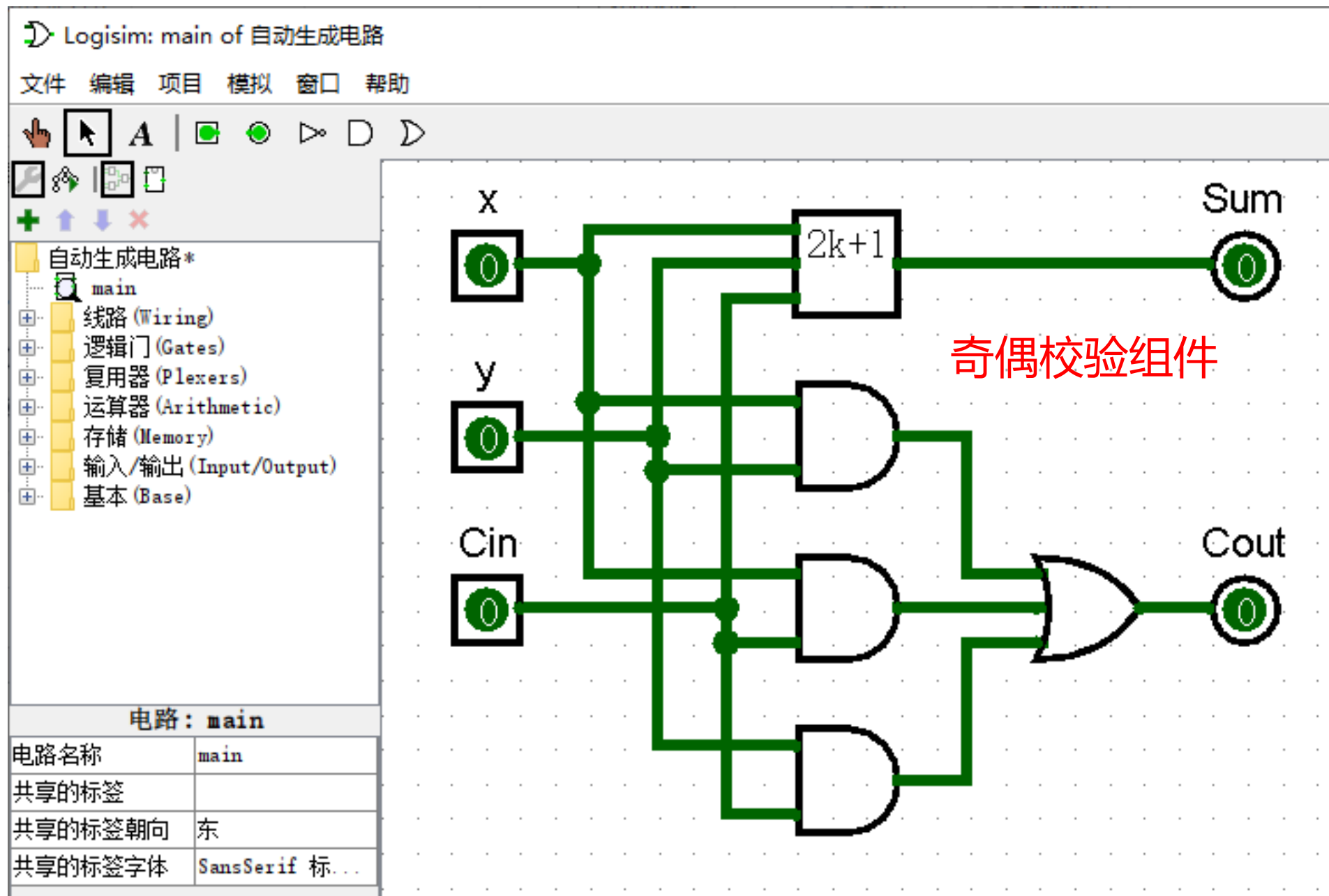
恢复

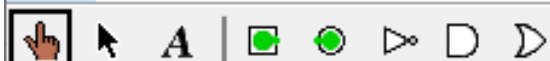
输入

建立电路









自动生成电路*

main

线路 (Wiring)

逻辑门 (Gates)

复用器 (Plexers)

运算器 (Arithmetic)

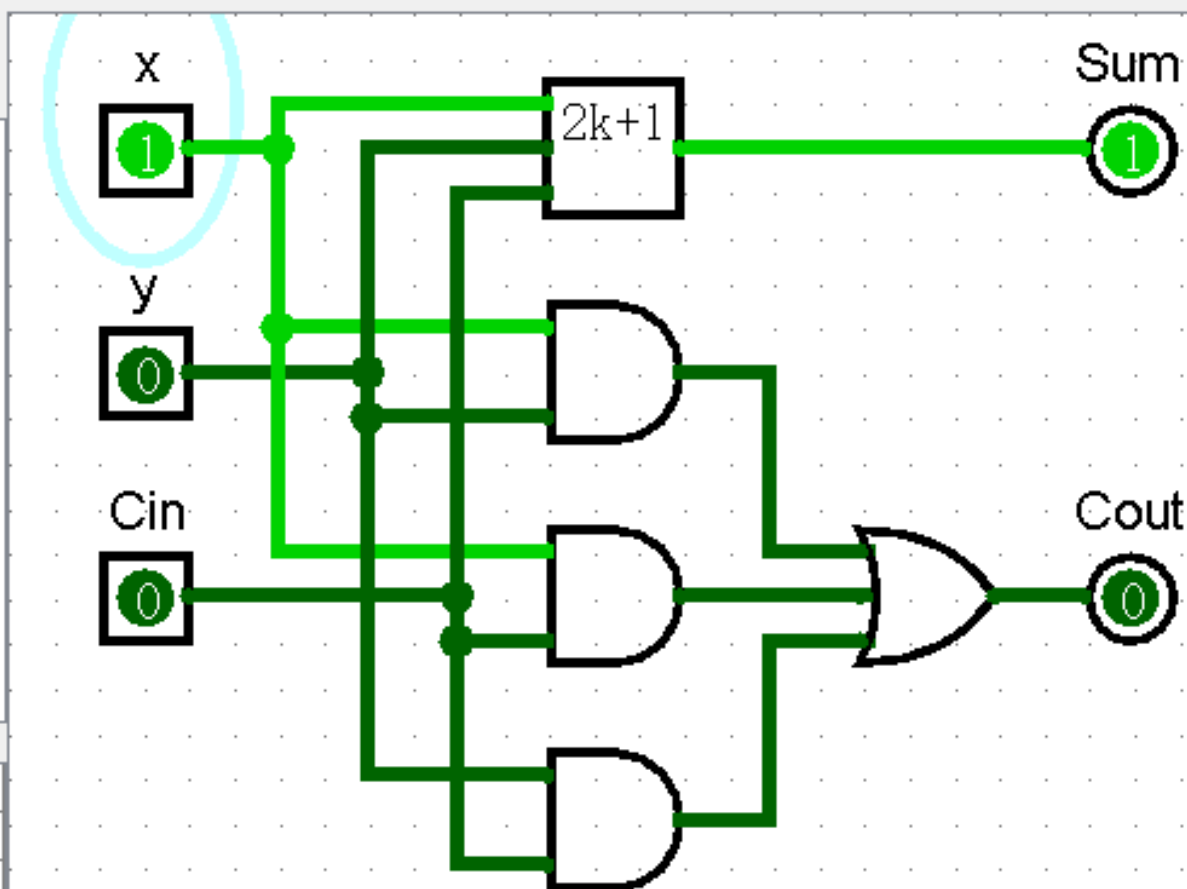
存储 (Memory)

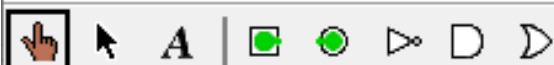
输入/输出 (Input/Output)

基本 (Base)

引脚 (Pin)

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	x
标签位置	北
标签字体	SansSerif 标...

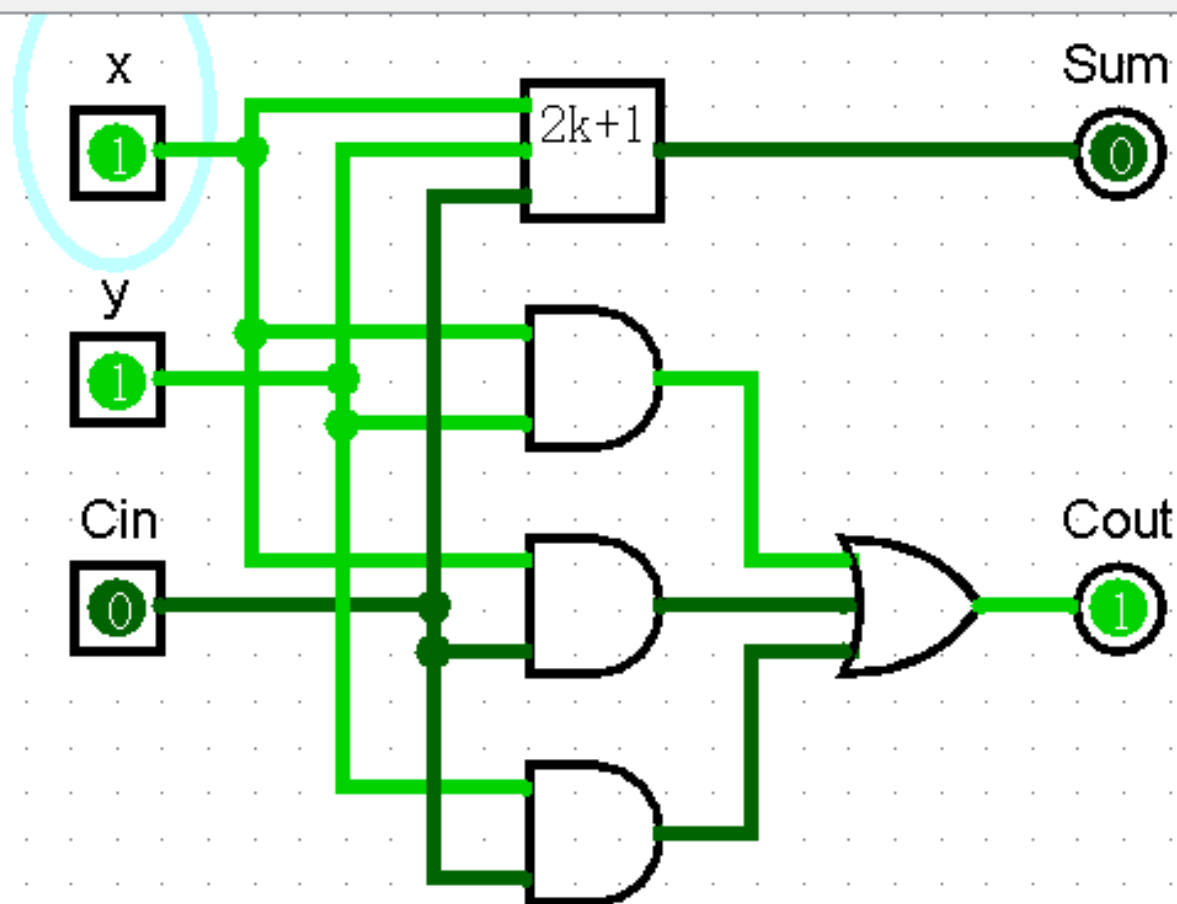


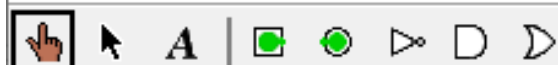


- 自动生成电路*
- main
- 线路 (Wiring)
- 逻辑门 (Gates)
- 复用器 (Plexers)
- 运算器 (Arithmetic)
- 存储 (Memory)
- 输入/输出 (Input/Output)
- 基本 (Base)

引脚 (Pin)

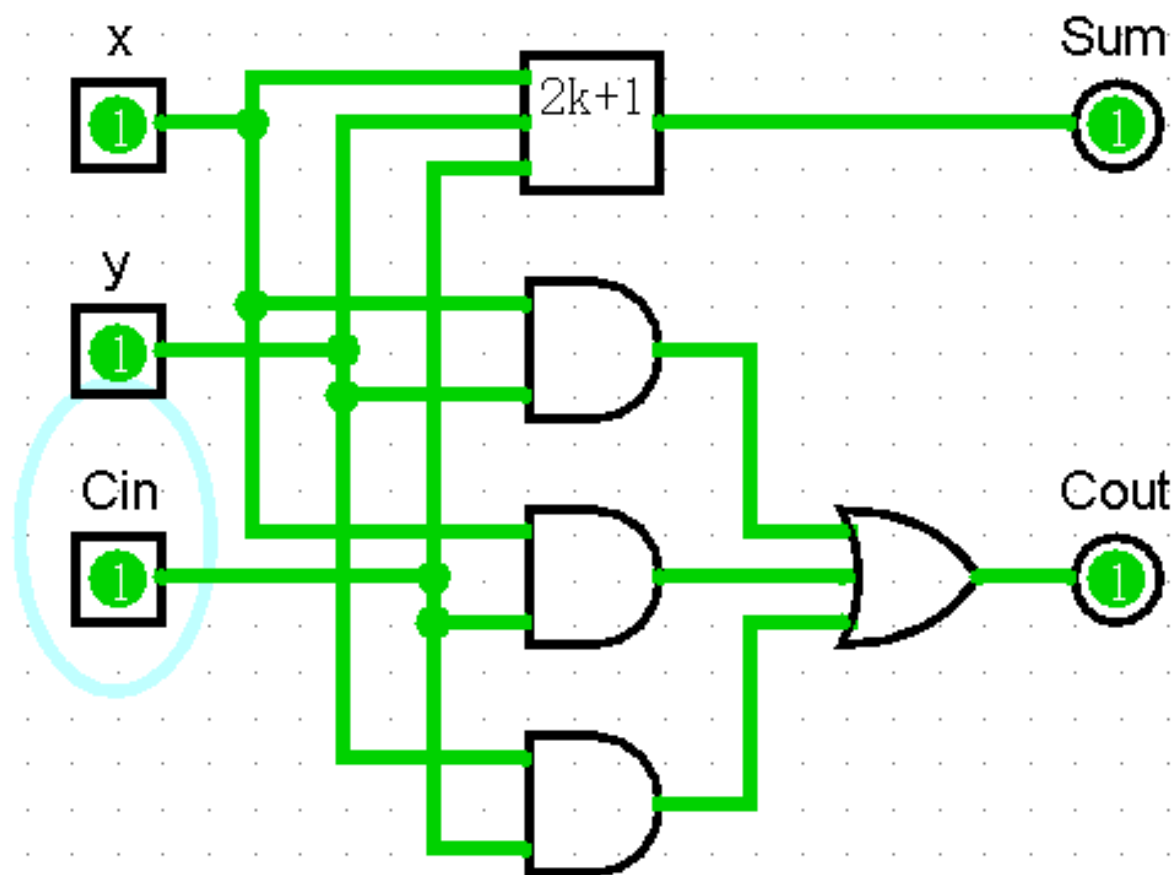
朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	x
标签位置	北
标签字体	SansSerif 标...





引脚 (Pin)

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	Cin
标签位置	北
标签字体	SansSerif 标...



Logisim: main of 自动生成电路

文件 编辑 项目 模拟 窗口 帮助

主工具栏: 选择、移动、放大、缩小、删除、复制、粘贴、撤销、重做、运行、暂停、单步、单步跳过、单步结束。

左侧树状图:

- 自动生成电路*
- main
- 线路 (Wiring)
- 逻辑门 (Gates)
- 复用器 (Plexers)
- 运算器 (Arithmetic)
- 存储 (Memory)
- 输入/输出 (Input/Output)
- 基本 (Base)

引脚 (Pin) 属性表:

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	Cin
标签位置	北
标签字体	SansSerif 标...

电路原理图:

Combinational Analysis 窗口:

文件 编辑 项目 模拟 窗口 帮助

输入 输出 真值表 表达式 最小项

表达式: $x y + x C$

建立电路 对话框:

目标项目: 自动生成电路

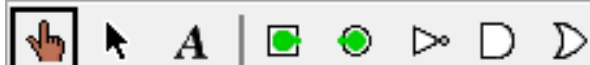
电路名称: main

☒ 仅用两个输入的逻辑门

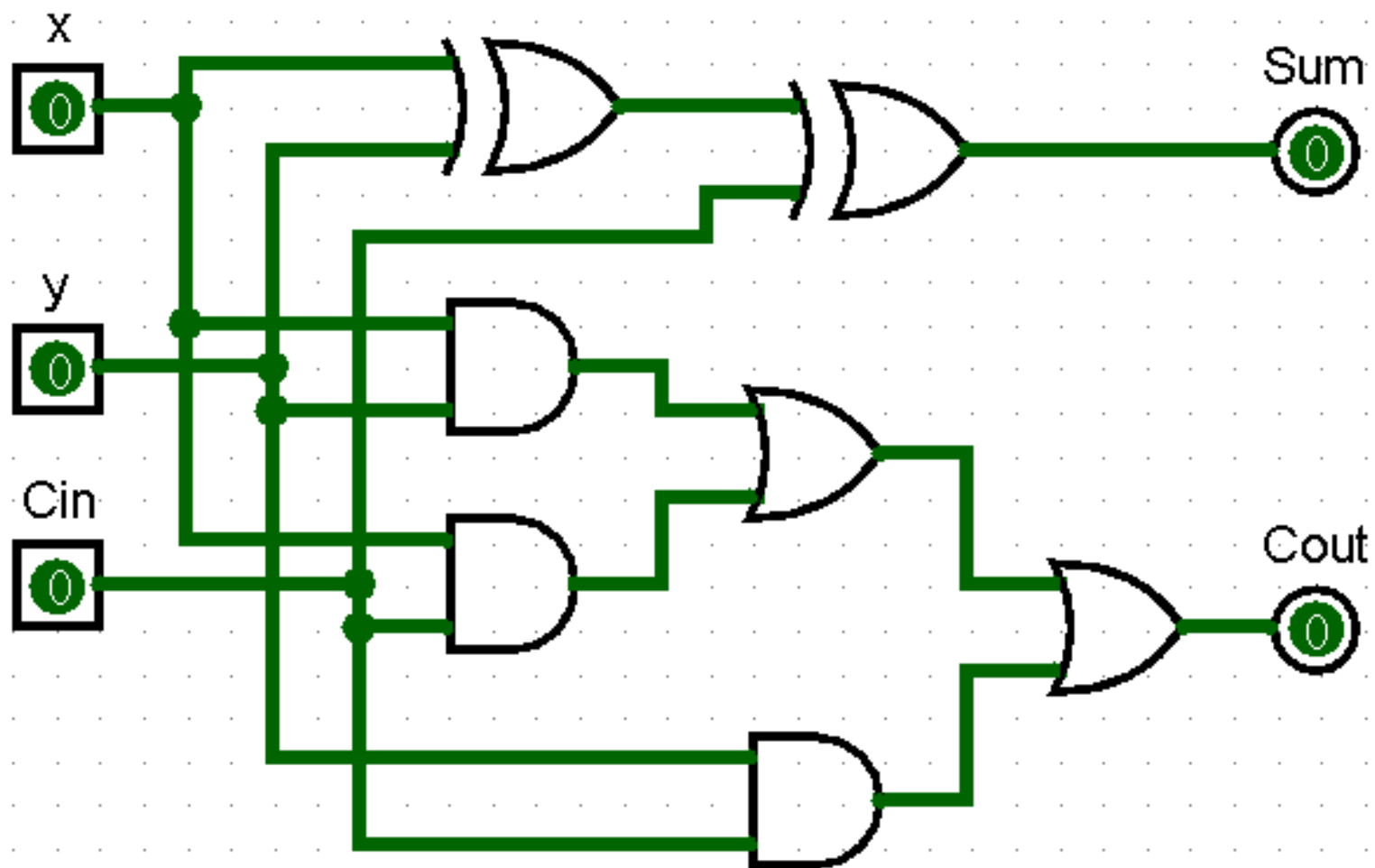
☐ 仅用与非 (NAND) 门

OK Cancel

建立电路



- 自动生成电路*
- main
- 线路 (Wiring)
- 逻辑门 (Gates)
- 复用器 (Plexers)
- 运算器 (Arithmetic)
- 存储 (Memory)
- 输入/输出 (Input/Output)
- 基本 (Base)

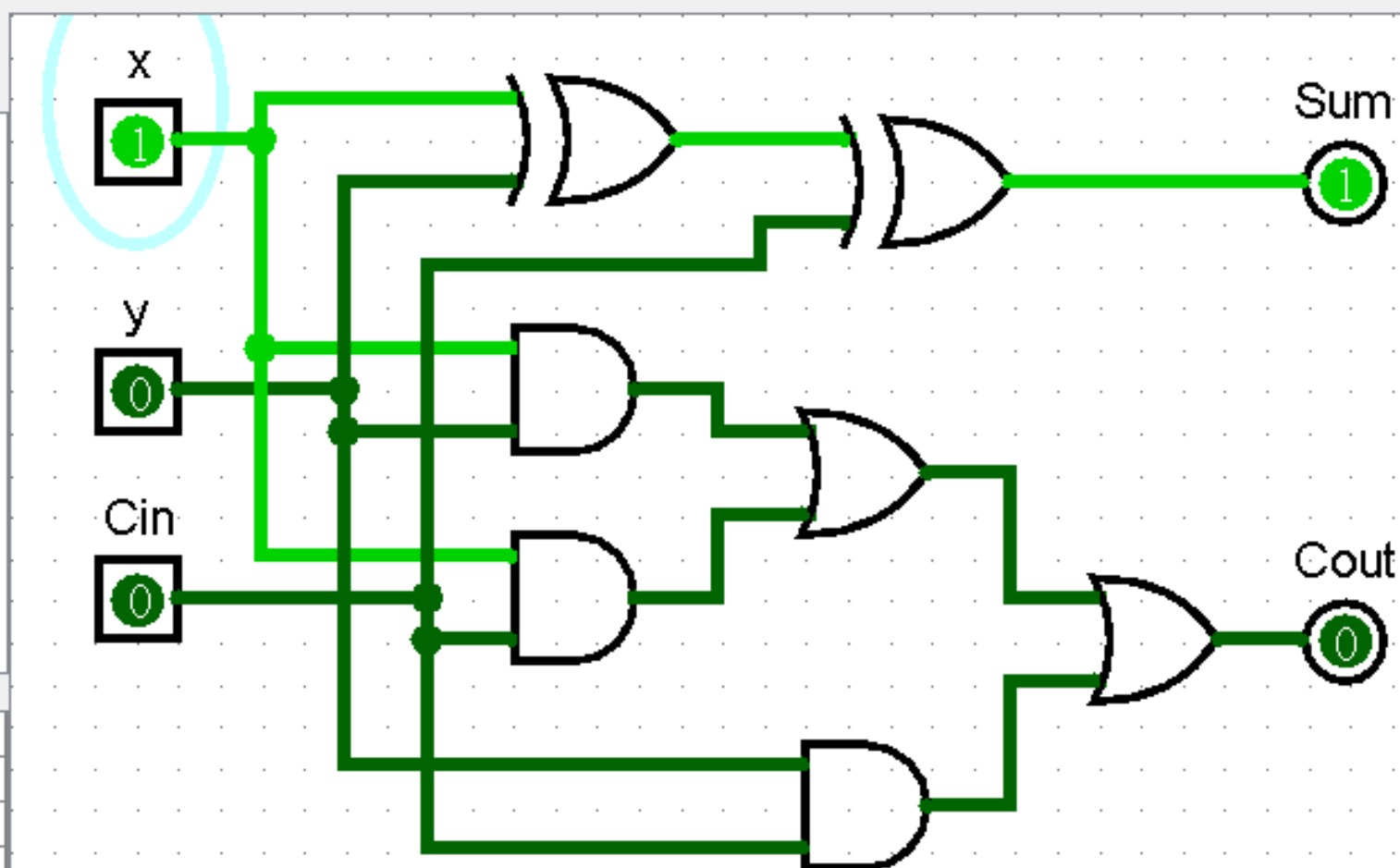




- 自动生成电路*
- main
- 线路 (Wiring)
- 逻辑门 (Gates)
- 复用器 (Plexers)
- 运算器 (Arithmetic)
- 存储 (Memory)
- 输入/输出 (Input/Output)
- 基本 (Base)

引脚 (Pin)

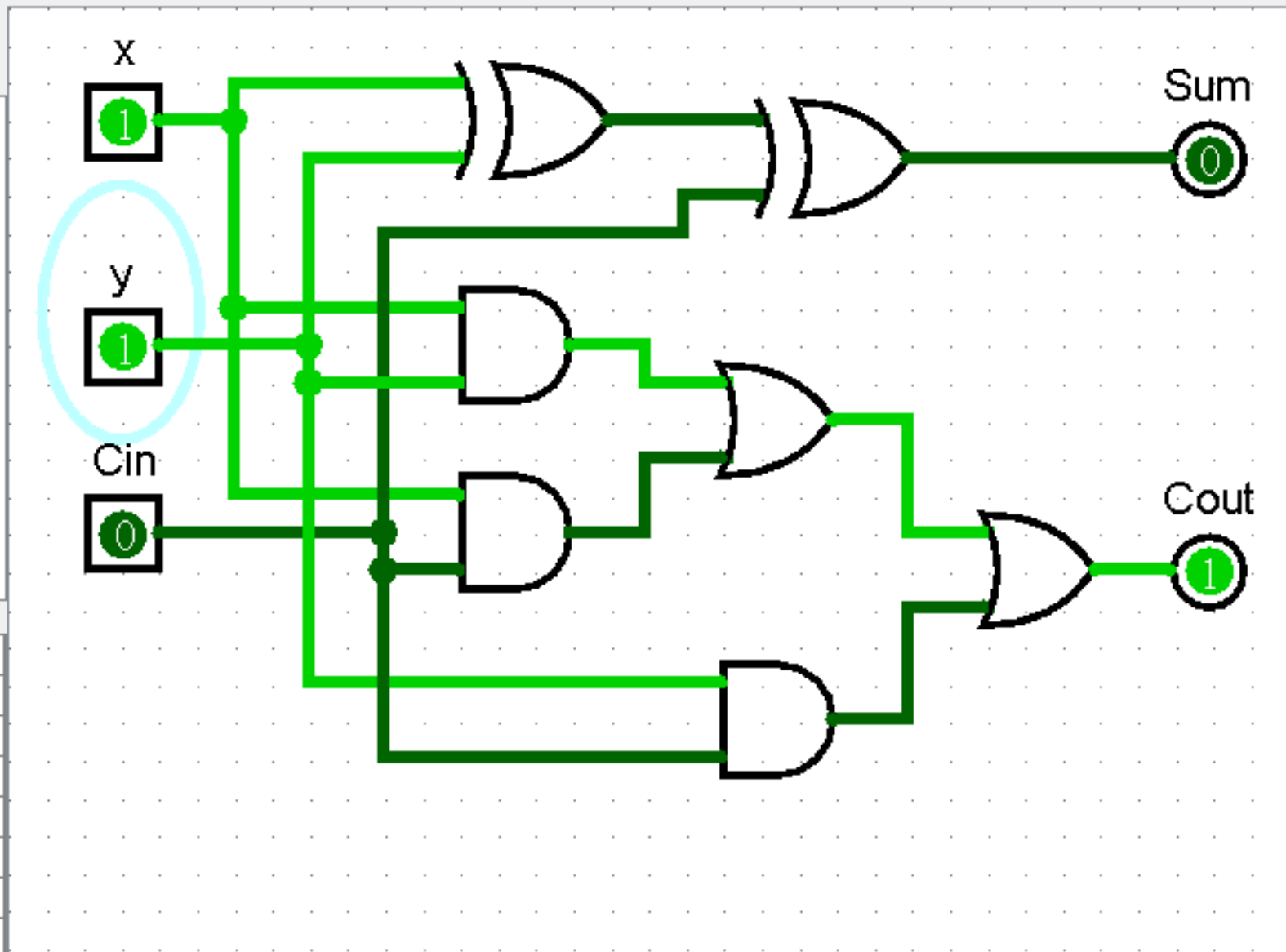
朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	x
标签位置	北
标签字体	SansSerif 标...

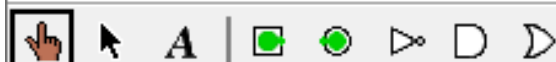




- 自动生成电路*
- main
- 线路 (Wiring)
- 逻辑门 (Gates)
- 复用器 (Plexers)
- 运算器 (Arithmetic)
- 存储 (Memory)
- 输入/输出 (Input/Output)
- 基本 (Base)

引脚 (Pin)	
朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	y
标签位置	北
标签字体	SansSerif 标...

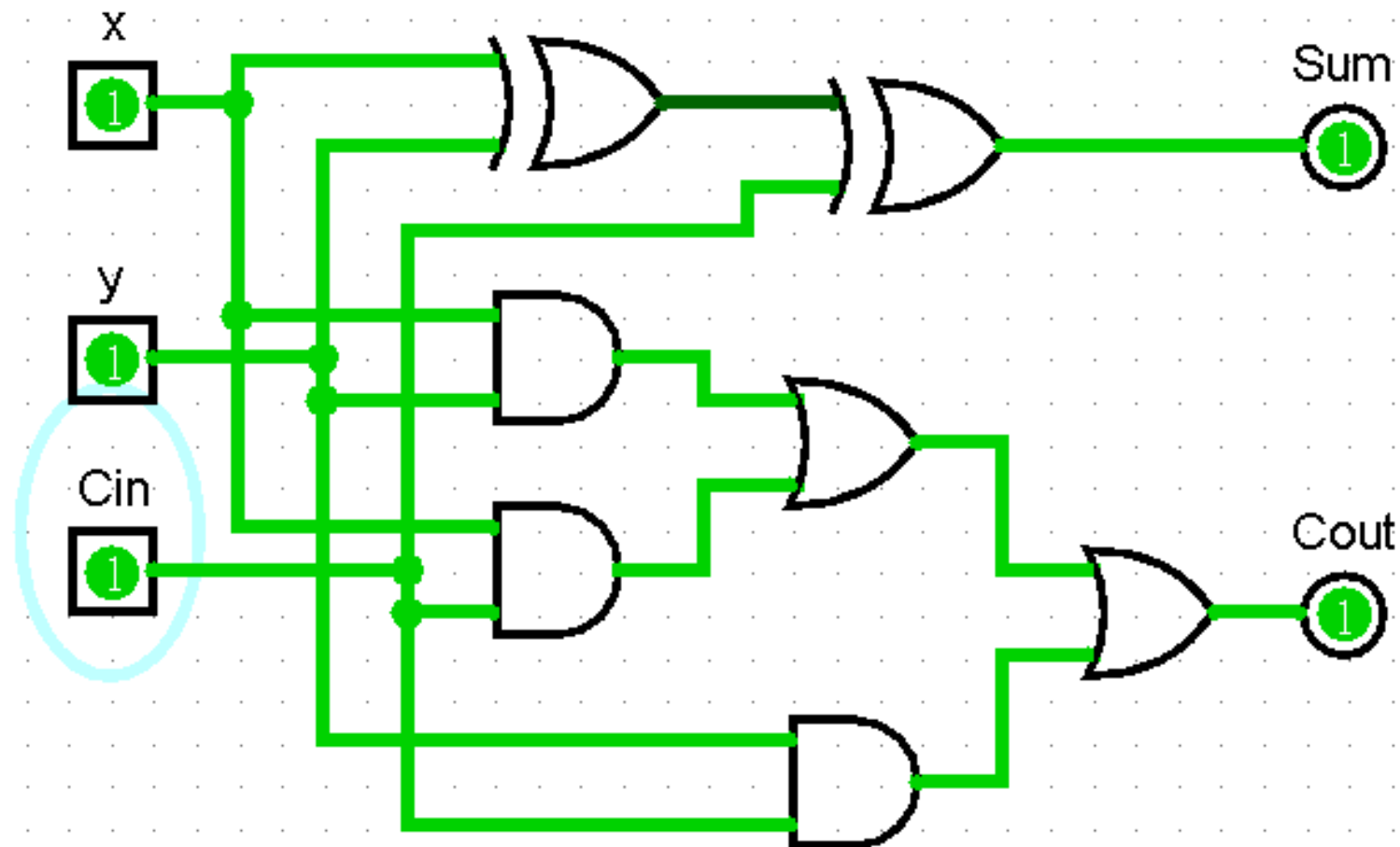




- 自动生成电路*
- main
- 线路 (Wiring)
- 逻辑门 (Gates)
- 复用器 (Plexers)
- 运算器 (Arithmetic)
- 存储 (Memory)
- 输入/输出 (Input/Output)
- 基本 (Base)

引脚 (Pin)

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	Cin
标签位置	北
标签字体	SansSerif 标...



■ 电路转化 (不用异或)

Logisim: main of 自动生成电路

文件 编辑 项目 模拟 窗口 帮助

自动生成电路*

- main
 - 线路 (Wiring)
 - 逻辑门 (Gates)
 - 复用器 (Plexers)
 - 运算器 (Arithmetic)
 - 存储 (Memory)
 - 输入/输出 (Input/Output)
 - 基本 (Base)

Combinational Analysis

文件 编辑 项目 模拟 窗口 帮助

输入 输出 真值表 表达式 最小项

格式: 积之和范式

y, Cin

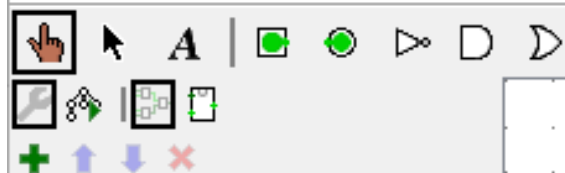
	00	01	11	10
x	0	1	0	1
y	1	0	1	0

卡诺图

$$\bar{x} \bar{y} \text{Cin} + \bar{x} y \text{Cin} + x \bar{y} \bar{\text{Cin}} + x y \text{Cin}$$

设为表达式

建立电路



自动生成电路

✚ 线路 (Wiring)

逻辑门 (Gates)

复用器 (Plexers)

运算器 (Arithmetic)

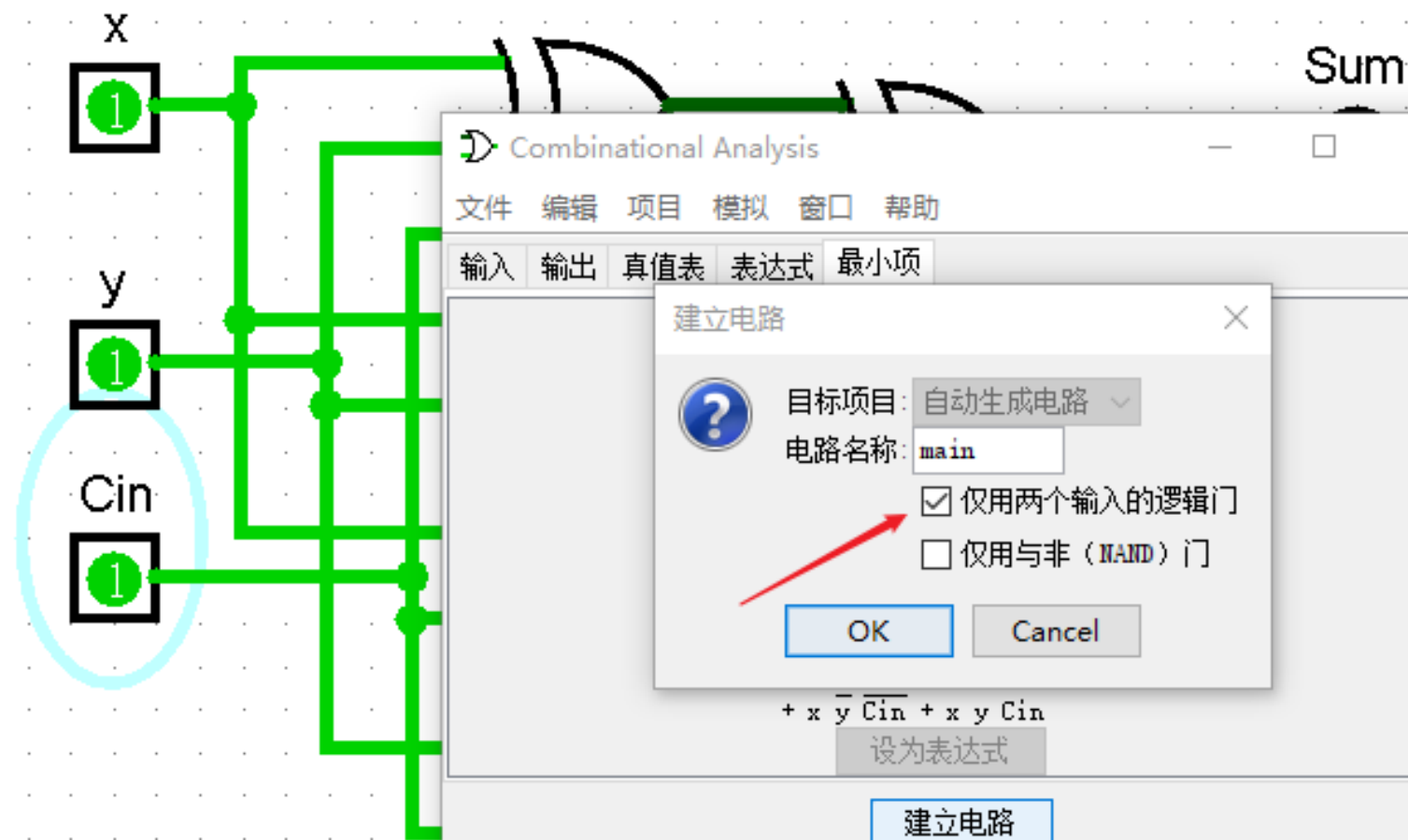
存储 (Memory)

输入/输出 (Input/Output)

基本 (Base)

引脚 (Pin)

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	Cin
标签位置	北
标签字体	SansSerif 标...



Combinational Analysis

文件 编辑 项目 模拟 窗口 帮助

输入	输出	真值表	表达式	最小项
----	----	-----	-----	-----

建立电路

目标项目: 自动生成电路

电路名称: main

☒ 仅用两个输入的逻辑门

☐ 仅用与非 (NAND) 门

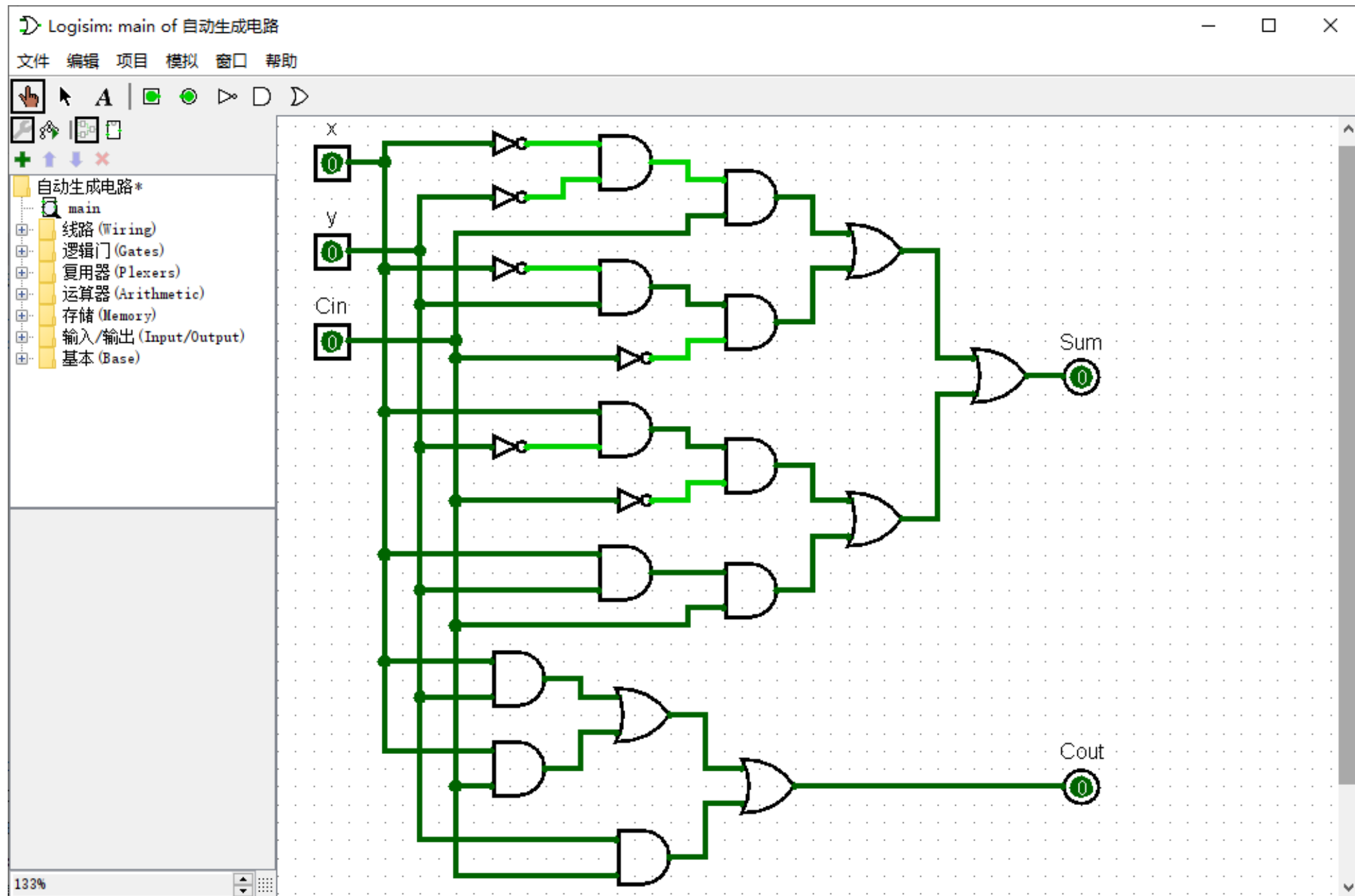
OK

Cancel

$$+ x \bar{y} \overline{C_{in}} + x y C_{in}$$

设为表达式

建立电路



仅有
与或非门



➤ Logisim中的延迟和险象

Logisim中的延迟和险象



■ Logisim中的组件有延迟吗？

- 是的
- 不能精准仿真不同器件的延迟
- 所有组件的延迟都是相同的
- 一个独立的组件，包含一级延迟 → 竞争

■ 竞争

- 在组合电路中，同一信号或同时变化的某些信号，经过不同路径到达某一点的时间有先有后，这种现象称为竞争
- 竞争 → 险象

■ 险象

- 由于竞争而引起电路输出发生瞬间错误的现象称为险象（冒险）
- 表现为输出端出现了原设计中没有的窄脉冲，常称为“毛刺”
- 在组合电路中，“毛刺”不一定造成严重后果
- 但当组合逻辑与时序逻辑结合在一起时，险象就可能造成严重错误，特别是当组合逻辑的输出作为时序电路的使能输入时

Logisim: 险象 of 险象

文件 编辑 项目 模拟 窗口 帮助

险象*

- 险象
- 震荡电路
- 线路 (Wiring)
- 逻辑门 (Gates)
- 复用器 (Plexers)
- 运算器 (Arithmetic)
 - 加法器 (Adder)
 - 减法器 (Subtract)
 - 乘法器 (Multipli)
 - 除法器 (Divider)
 - 补码器 (Negator)
 - 比较器 (Comparat)
 - 移位器 (Shifter)

引脚 (Pin)

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	
标签位置	西
标签字体	SansSerif ...

输出恒为0

Logisim: 险象 of 险象

文件 编辑 项目 模拟 窗口 帮助

险象*

- 险象
- 震荡电路
- 线路 (Wiring)
- 逻辑门 (Gates)
- 复用器 (Plexers)
- 运算器 (Arithmetic)
 - 加法器 (Adder)
 - 减法器 (Subtractor)
 - 乘法器 (Multiplier)
 - 除法器 (Divider)
 - 补码器 (Negator)
 - 比较器 (Comparator)
 - 移位器 (Shifter)

引脚 (Pin)

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	
标签位置	西
标签字体	SansSerif

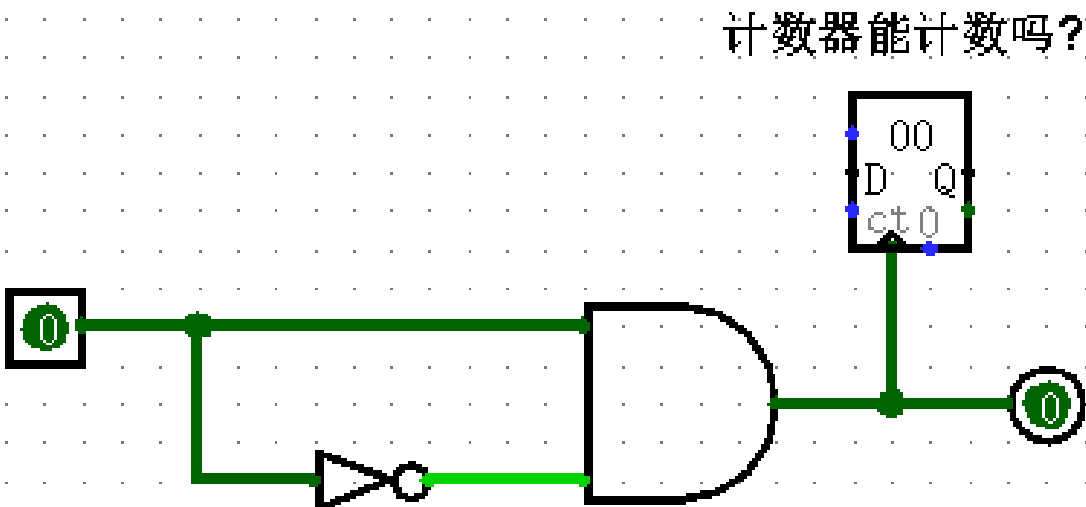
输出恒为0



- ? 位查找器 (Bit Fi ^
- 存储 (Memory)
 - D 触发器 (D Flip-
 - T 触发器 (T Flip-
 - JK 触发器 (J-K Fl
 - SR 锁存器 (S-R Fl
 - 寄存器 (Register
 - 计数器 (Counter)
 - 移位寄存器 (Shif
 - 随机数生成器 (Ra
 - 随机存储器 (RAM)
 - 只读存储器 (ROM)
- 输入/输出 (Input/Out
- 基本 (Base)

电路：险象

电路名称	险象
共享的标签	
共享的标签...	东
共享的标签...	SansSerif ...



输出恒为0
计数器应该
不能计数

Logisim: 险象 of 险象

文件 编辑 项目 模拟 窗口 帮助

险象*

- 险象
- 振荡电路
- 线路 (Wiring)
- 逻辑门 (Gates)
- 复用器 (Plexers)
- 运算器 (Arithmetic)
 - 加法器 (Adder)
 - 减法器 (Subtract)
 - 乘法器 (Multipli
 - 除法器 (Divider)
 - 补码器 (Negator)
 - 比较器 (Comparat
 - 移位器 (Shifter)

电路: 险象

电路名称	险象
共享的标签	
共享的标签...	东
共享的标签...	SansSerif ...

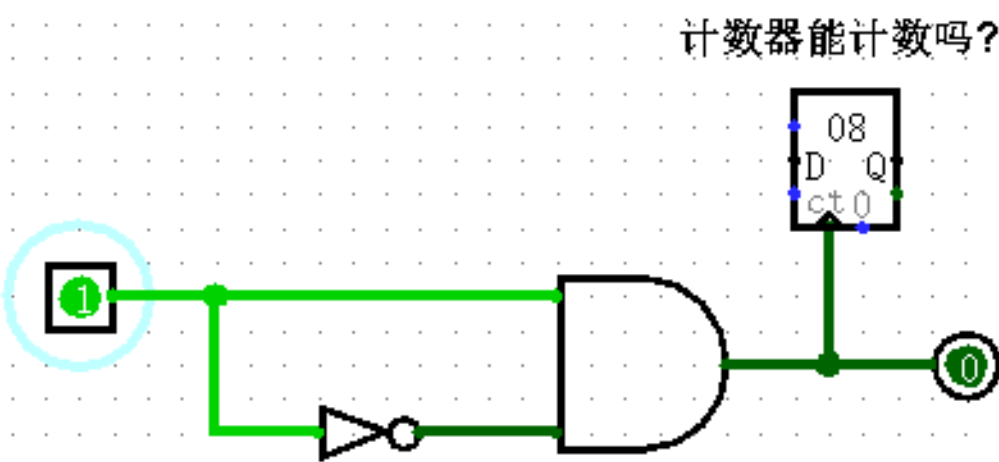
计数器能计数吗?

存在时钟信号



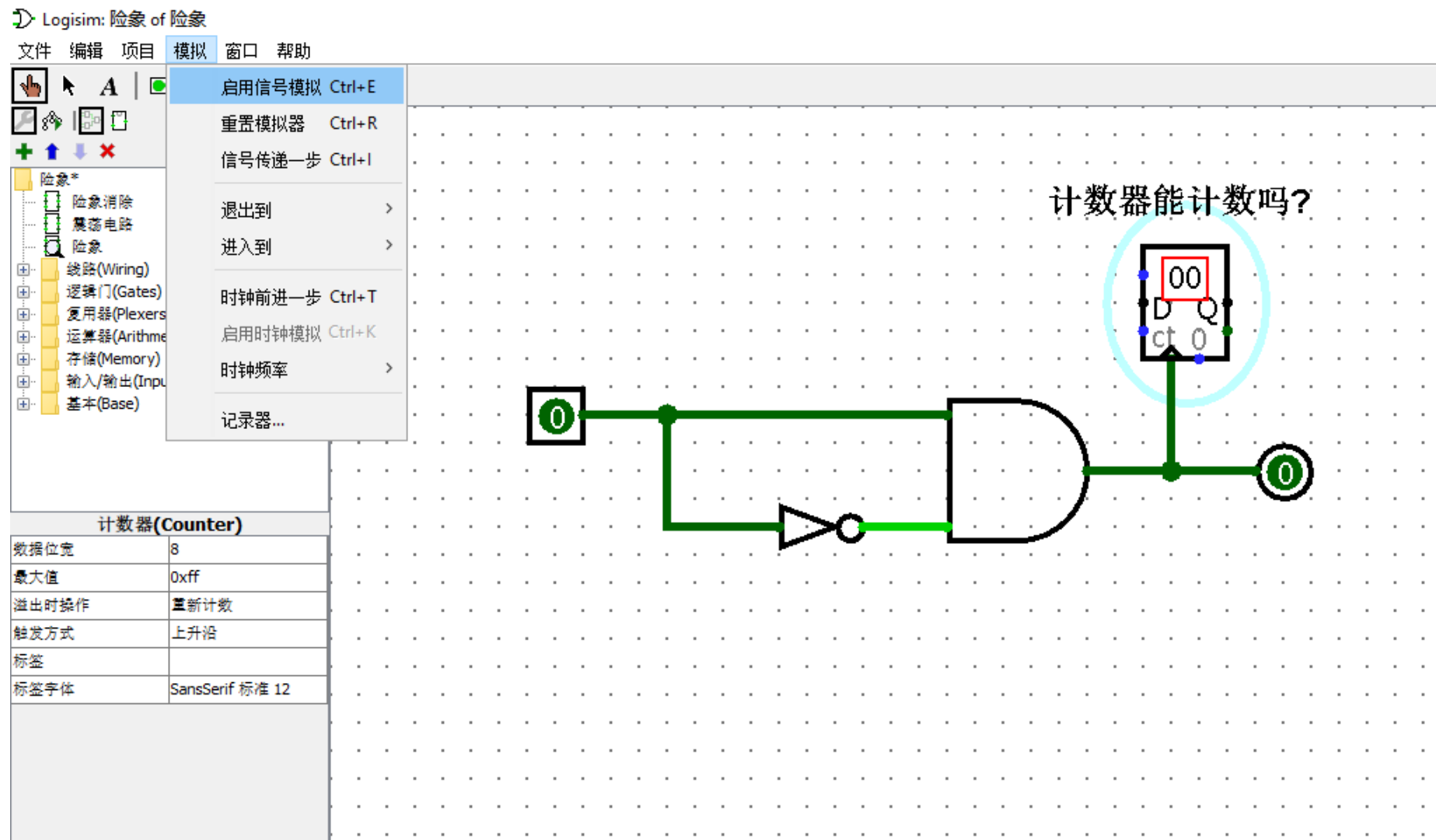
- 位查找器 (Bit Fi)
- 存储 (Memory)
 - D触发器 (D Flip-
 - T触发器 (T Flip-
 - JK触发器 (J-K Fl
 - RS锁存器 (S-R Fl
 - 寄存器 (Register
 - 计数器 (Counter)
 - 移位寄存器 (Shif
 - 随机数生成器 (Ra
 - 随机存储器 (RAM)
 - 只读存储器 (ROM)
- 输入/输出 (Input/Out
- 基本 (Base)

引脚 (Pin)	
朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	
标签位置	西
标签字体	SansSerif ...



存在时钟信号

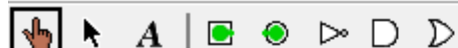
- 关闭持续启用信号模拟，采用单步信号传递
- 利用手形工具点击计数器的计数，可直接输入数字进行清零



■ 改变输入引脚值

Logisim: 险象 of 险象

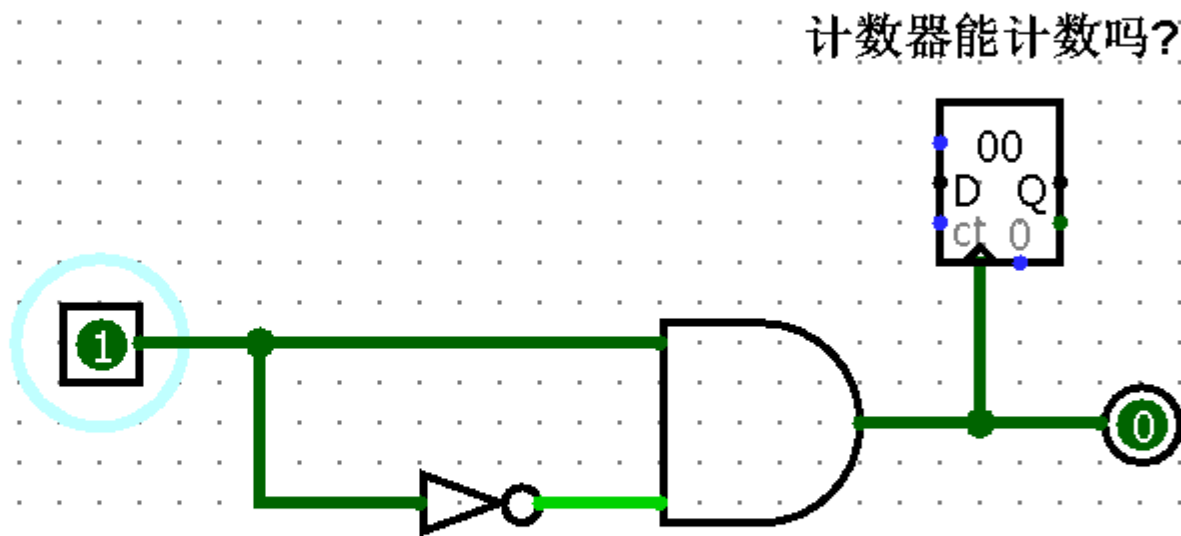
文件 编辑 项目 模拟 窗口 帮助



- 险象*
- 险象消除
- 震荡电路
- 险象
- 线路(Wiring)
- 逻辑门(Gates)
- 复用器(Multiplexers)
- 运算器(Arithmetic)
- 存储器(Memory)
- 输入/输出(Input/Output)
- 基本(Base)

引脚(Pin)

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	
标签位置	西
标签字体	SansSerif 标准 12



Logisim: 险象 of 险象

文件 编辑 项目 模拟 窗口 帮助

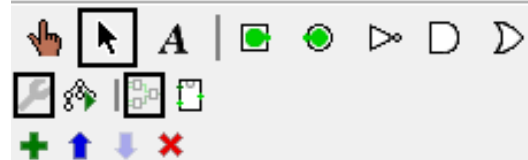
启用信号模拟 Ctrl+E
重置模拟器 Ctrl+R
信号传递一步 Ctrl+I
退出到 >
进入到 >
时钟前进一步 Ctrl+T
启用时钟模拟 Ctrl+K
时钟频率 >
记录器...

位查找器
存储 (Memory)
D 触发器 (D)
T 触发器 (T)
JK 触发器 (JK)
RS 锁存器 (SR)
寄存器 (R)
计数器 (C)
移位寄存器 (S)
随机数生成器 (RNG)
随机存储 (RAM)
只读存储 (ROM)
输入/输出 (Input/Output)
基本 (Base)

引脚 (Pin)

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	
标签位置	西
标签字体	SansSerif ...

计数器能计数吗?



险象*

- 险象消除
- 震荡电路
- 险象

线路(Wiring)

逻辑门(Gates)

复用器(Plexers)

运算器(Arithmetic)

存储(Memory)

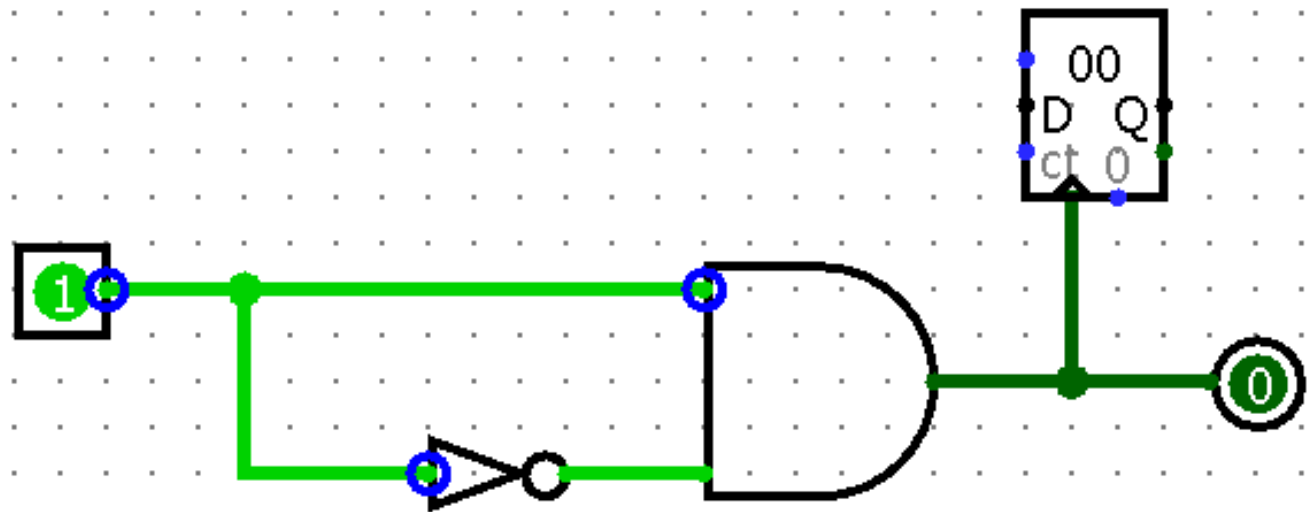
输入/输出(Input/Output)

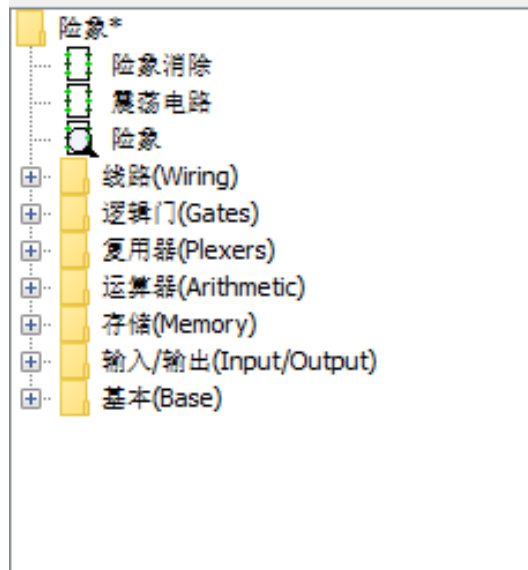
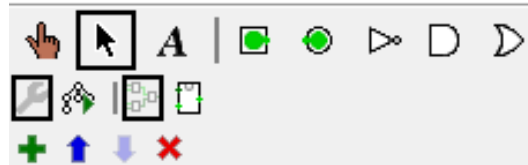
基本(Base)

电路: 险象

电路名称	险象
共享的标签	
共享的标签朝向	东
共享的标签字体	SansSerif 标准 12

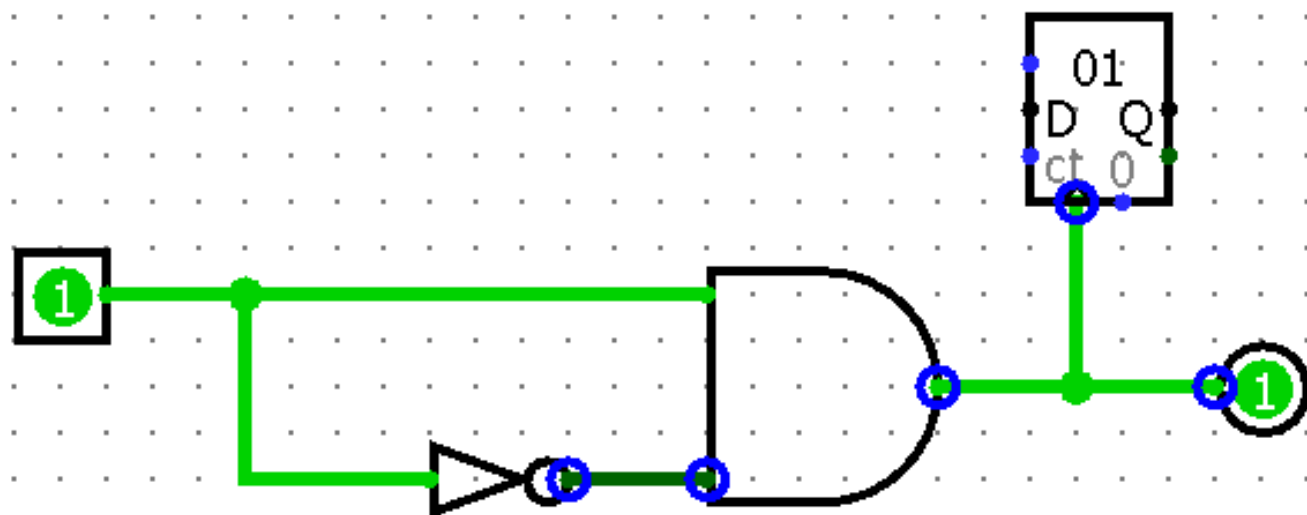
计数器能计数吗?

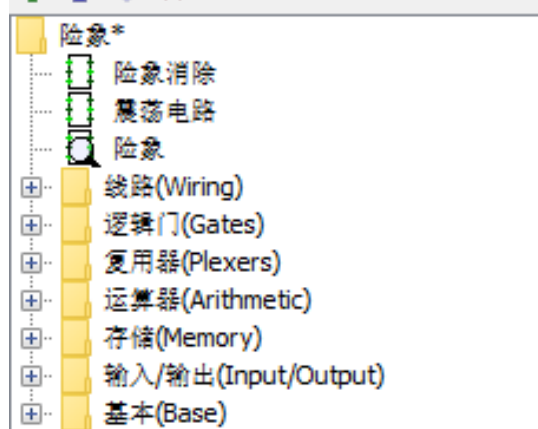
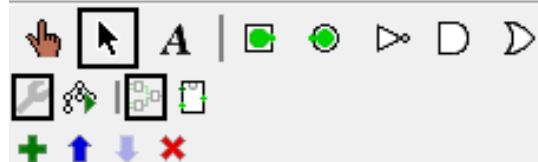




电路: 险象	
电路名称	险象
共享的标签	
共享的标签朝向	东
共享的标签字体	SansSerif 标准 12

计数器能计数吗?

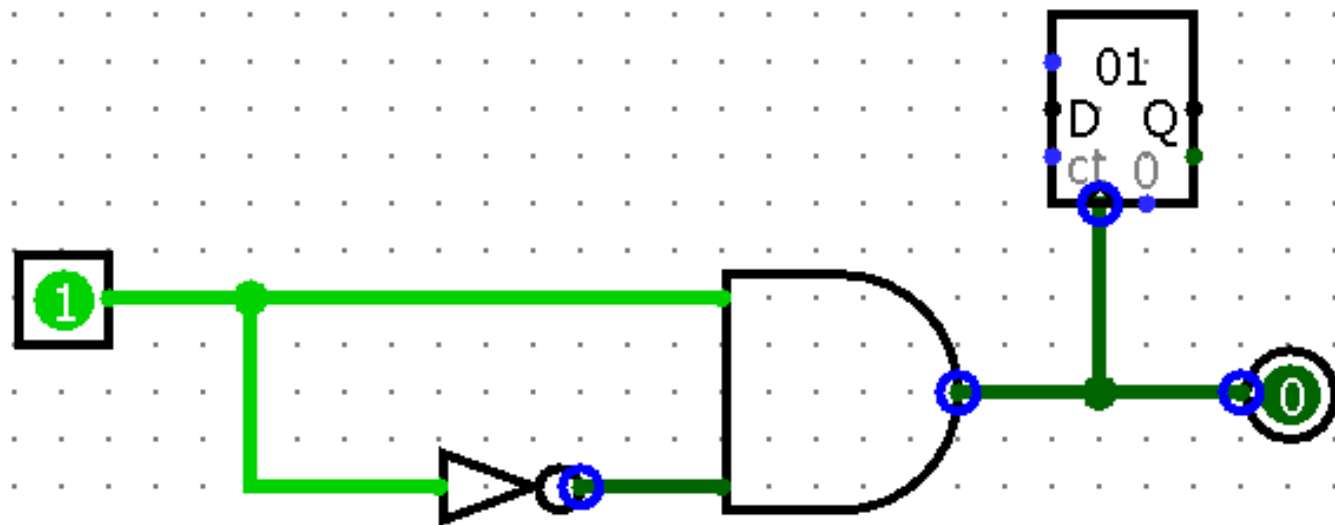




电路: 险象

电路名称	险象
共享的标签	
共享的标签朝向	东
共享的标签字体	SansSerif 标准 12

计数器能计数吗?

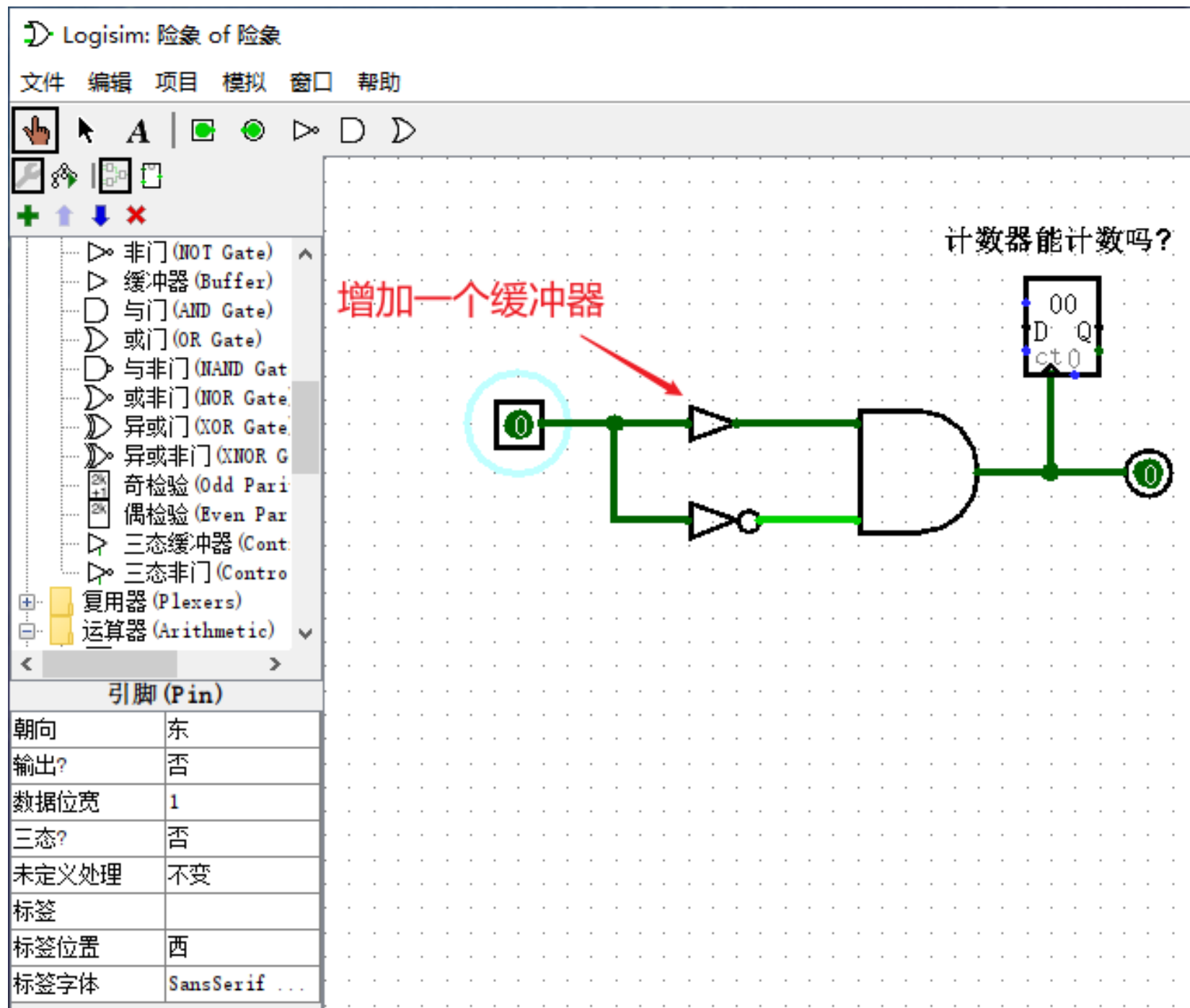


■ 消除竞争

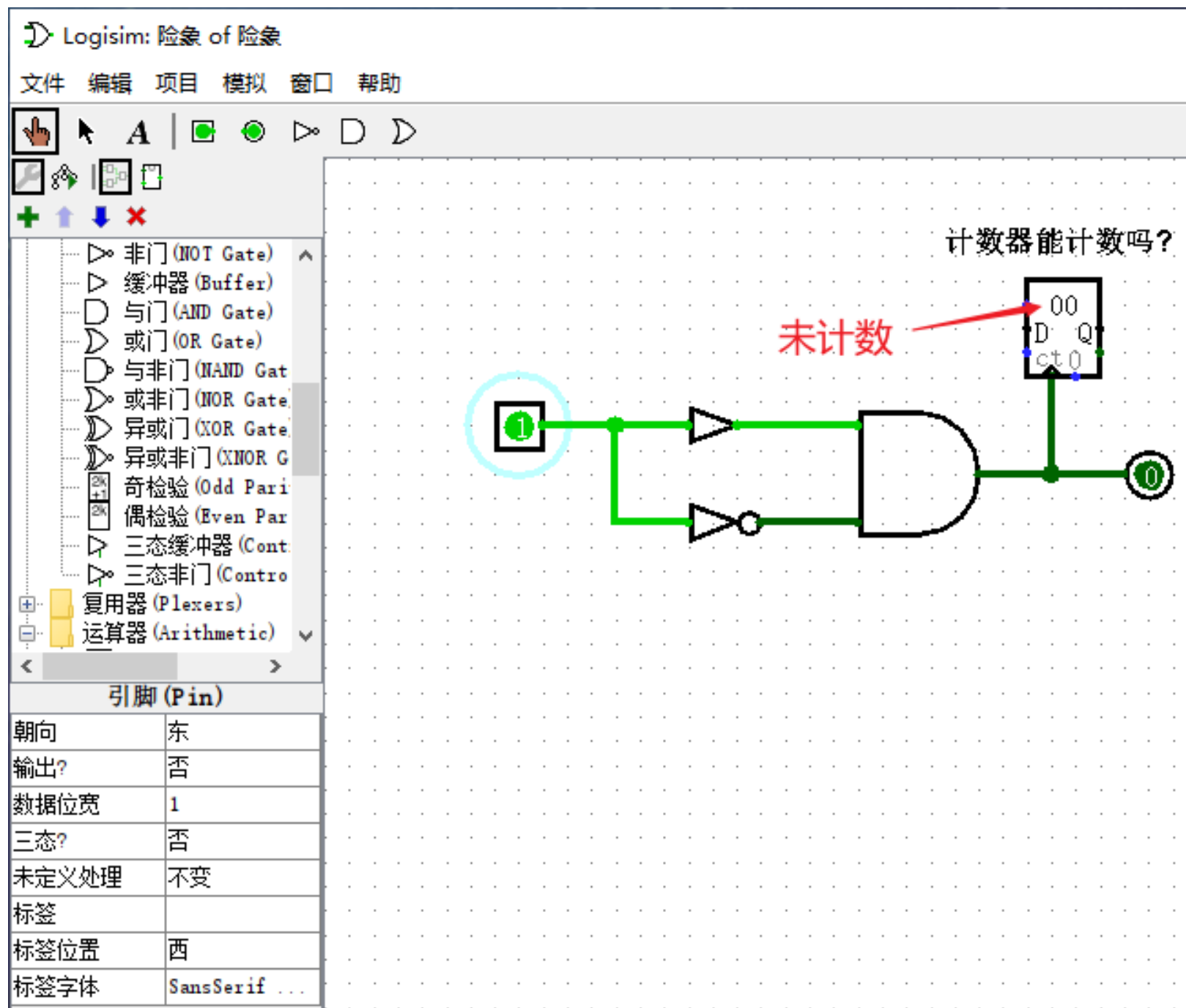
- 增加一些缓冲器使得各个输入端的信号的传播路径一致

■ 缓冲器

- 起到一级延迟的作用



- 持续启用信号模拟
- 复位计数器的值
- 改变输入值
 - 计数器不计数



- 非门直接集成到与门，
成为一个组件
- 输入信号传播路径一
致

Logisim: 险象 of 险象

文件 编辑 项目 模拟 窗口 帮助

工具栏: 选择, 放置, 删除, 复制, 粘贴, 撤销, 重做, 运行, 暂停, 单步, 单步退出, 帮助

组件列表:

- 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)
- 异或非门 (XNOR Gate)
- 奇检验 (Odd Parity)
- 偶检验 (Even Parity)
- 三态缓冲器 (Tri-state Buffer)
- 三态非门 (Tri-state Inverter)
- 复用器 (Multiplexers)
- 运算器 (Arithmetic)

选区: 与门 (AND Gate)

朝向	东
数据位宽	1
门尺寸	中等
输入引脚数量	2
输出值	0/1
标签	
标签字体	SansSerif ...
反转1 (顶部)	否
反转2 (底部)	是

计数器能计数吗?

修改与门的另外一个引脚



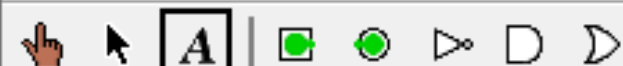
➤ Logisim中的震荡现象

Logisim中的震荡现象



■ 震荡现象

- 电路由于不恰当反馈回路的存在，无法进入稳态
- Logisim的仿真算法，会陷入死循环
- 系统的保护机制：循环次数达到事先设定的阈值的时候，系统就判定当前电路存在着震荡，从而退出自动仿真，避免系统仿真算法陷入死循环

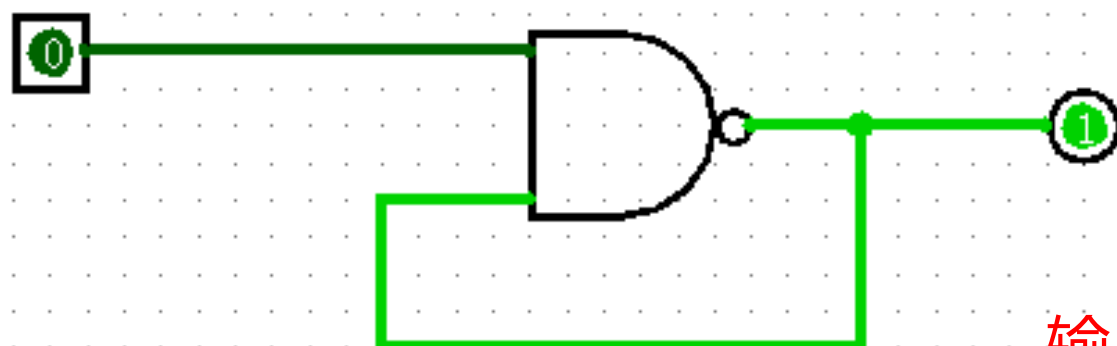


- 线路 (Wiring)
- 逻辑门 (Gates)
 - 非门 (NOT Gate)
 - 缓冲器 (Buffer)
 - 与门 (AND Gate)
 - 或门 (OR Gate)
 - 与非门 (NAND Gate)
 - 或非门 (NOR Gate)
 - 异或门 (XOR Gate)
 - 异或非门 (XNOR Gate)
 - 奇检验 (Odd Parity)
 - 偶检验 (Even Parity)
 - 三态缓冲器 (Tri-state Buffer)
 - 三态非门 (Tri-state Inverter)



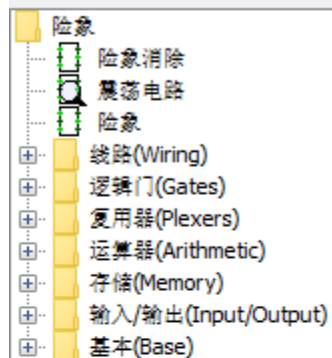
标签 (Label)

文本	
字体	SansSerif ...
水平对齐 (...)	中心
竖直对齐 (...)	基线



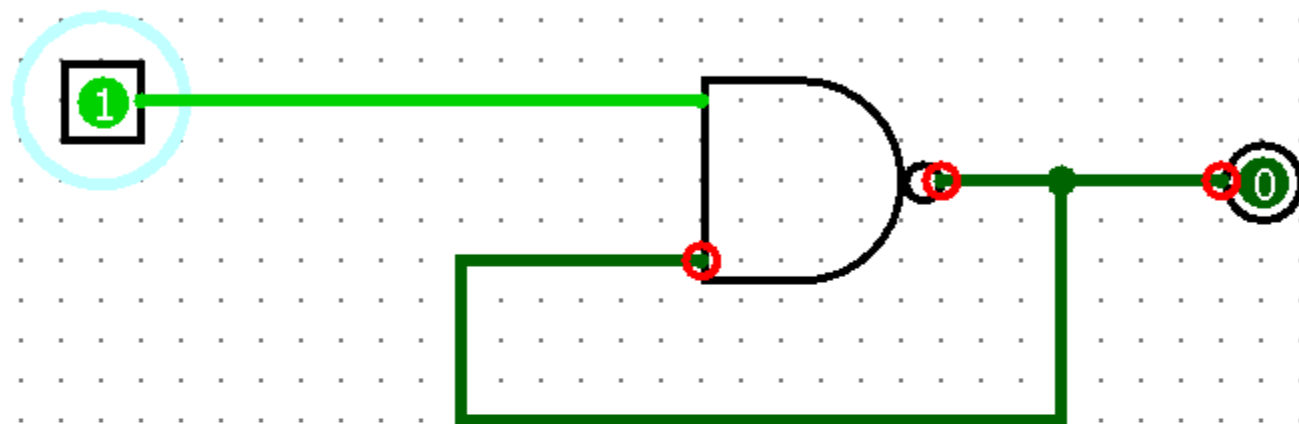
反馈回路

输入为0, 稳定状态



引脚 (Pin)

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	
标签位置	西
标签字体	SansSerif 标准 12



反馈回路

输入为1，震荡状态

存在明显的震荡



启用信号模拟 Ctrl+E

重置模拟器 Ctrl+R

信号传递一步 Ctrl+I

退出到

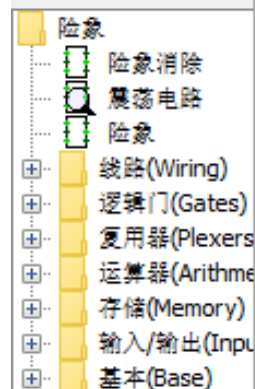
进入到

时钟前进一步 Ctrl+T

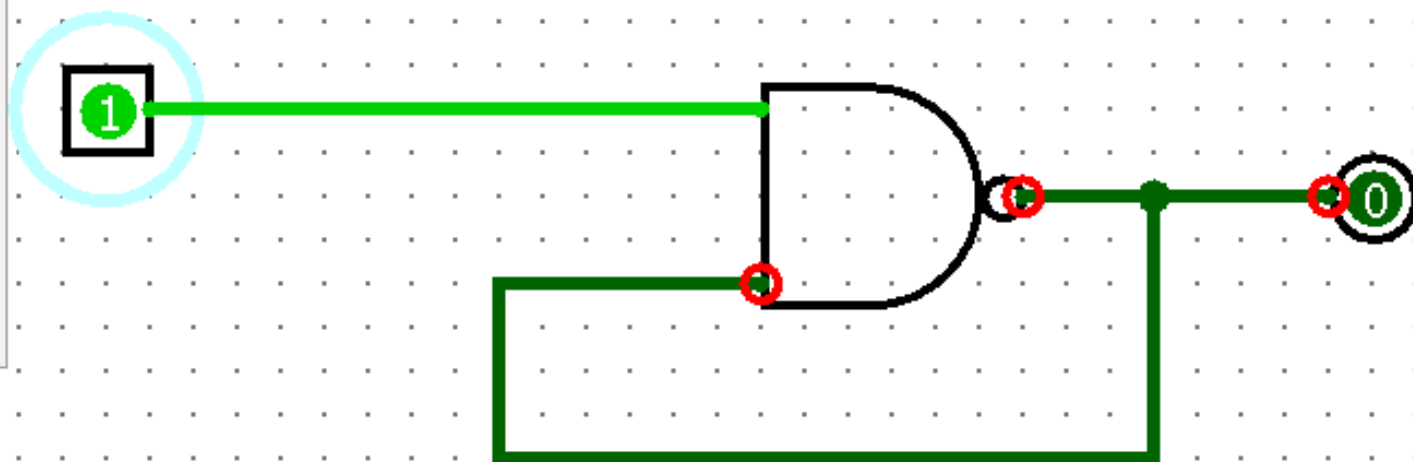
启用时钟模拟 Ctrl+K

时钟频率

记录器...



引脚 (Pin)	
朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	
标签位置	西
标签字体	SansSerif 标准 12



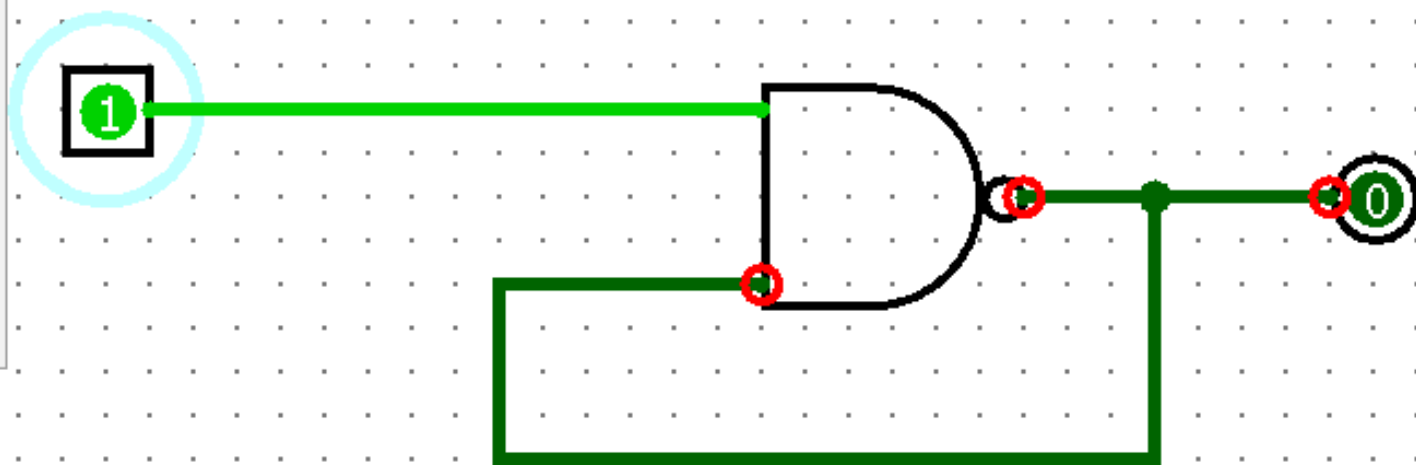
反馈回路

自动退出持续信号
模拟

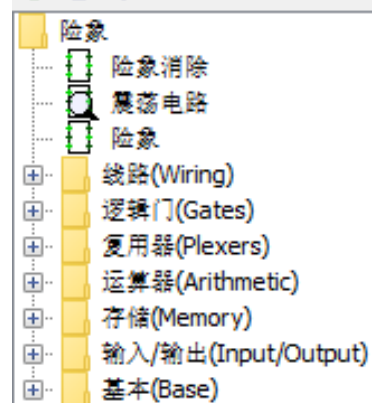
Logisim 工具栏和菜单:

- 工具栏: 选择、移动、放大、缩小、删除、复制、粘贴、撤销、重做、总线、输入、输出、时钟、探针、记录器、信号源、多路复用器、门、寄存器、触发器、计数器、显示、总线分析仪、信号发生器、时钟发生器、频率计、示波器、逻辑分析仪、信号发生器、时钟发生器、频率计、示波器、逻辑分析仪。
- 菜单: 启用信号模拟 (Ctrl+E)、重置模拟器 (Ctrl+R)、信号传递一步 (Ctrl+I)、退出到、进入到、时钟前进一步 (Ctrl+T)、启用时钟模拟 (Ctrl+K)、时钟频率、记录器...

引脚 (Pin)	
朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	
标签位置	西
标签字体	SansSerif 标准 12

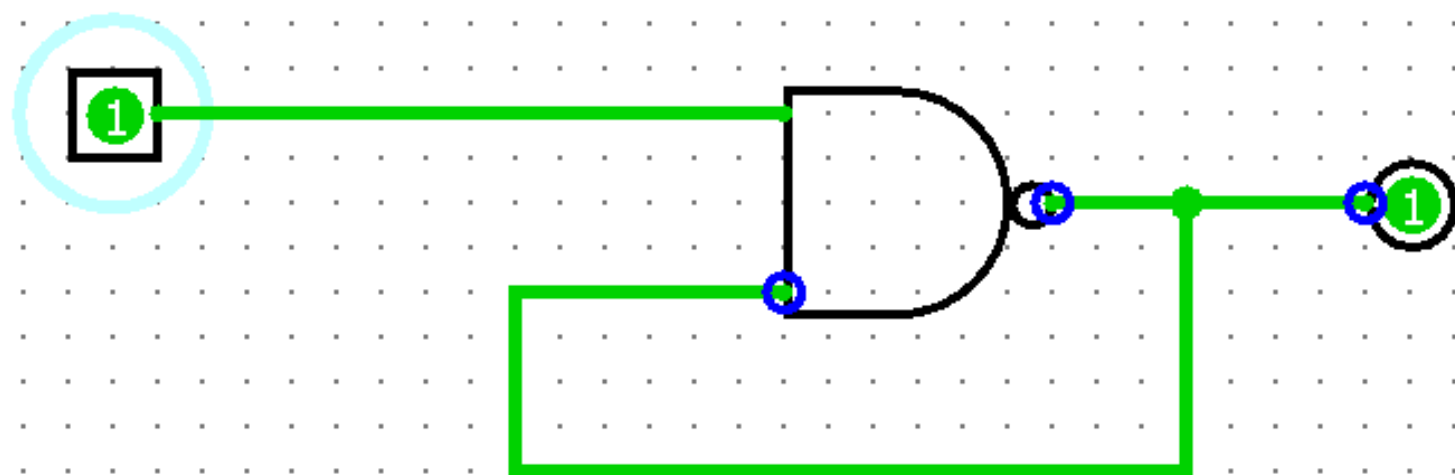


反馈回路



引脚(Pin)

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	
标签位置	西
标签字体	SansSerif 标准 12



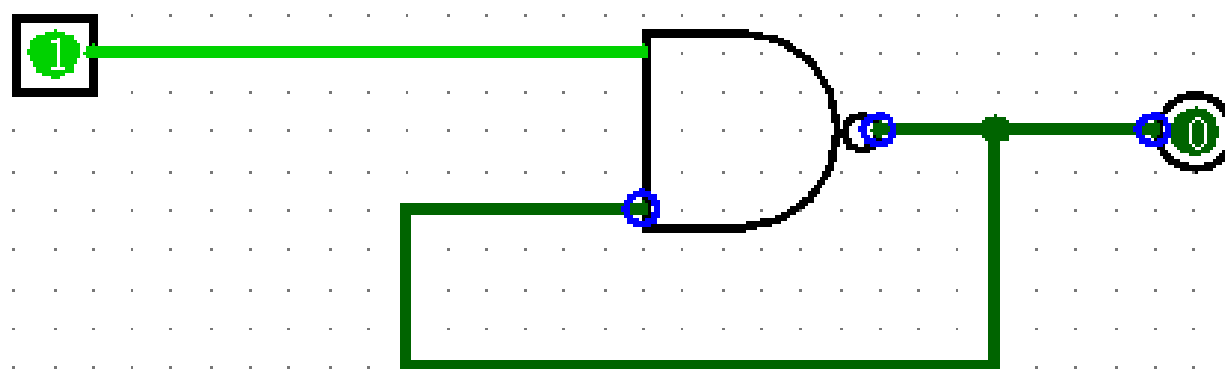
反馈回路



- 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)
- 异或非门 (XNOR Gate)

电路：震荡电路

电路名称	震荡电路
共享的标签	
共享的标签...	东
共享的标签...	SansSerif ...

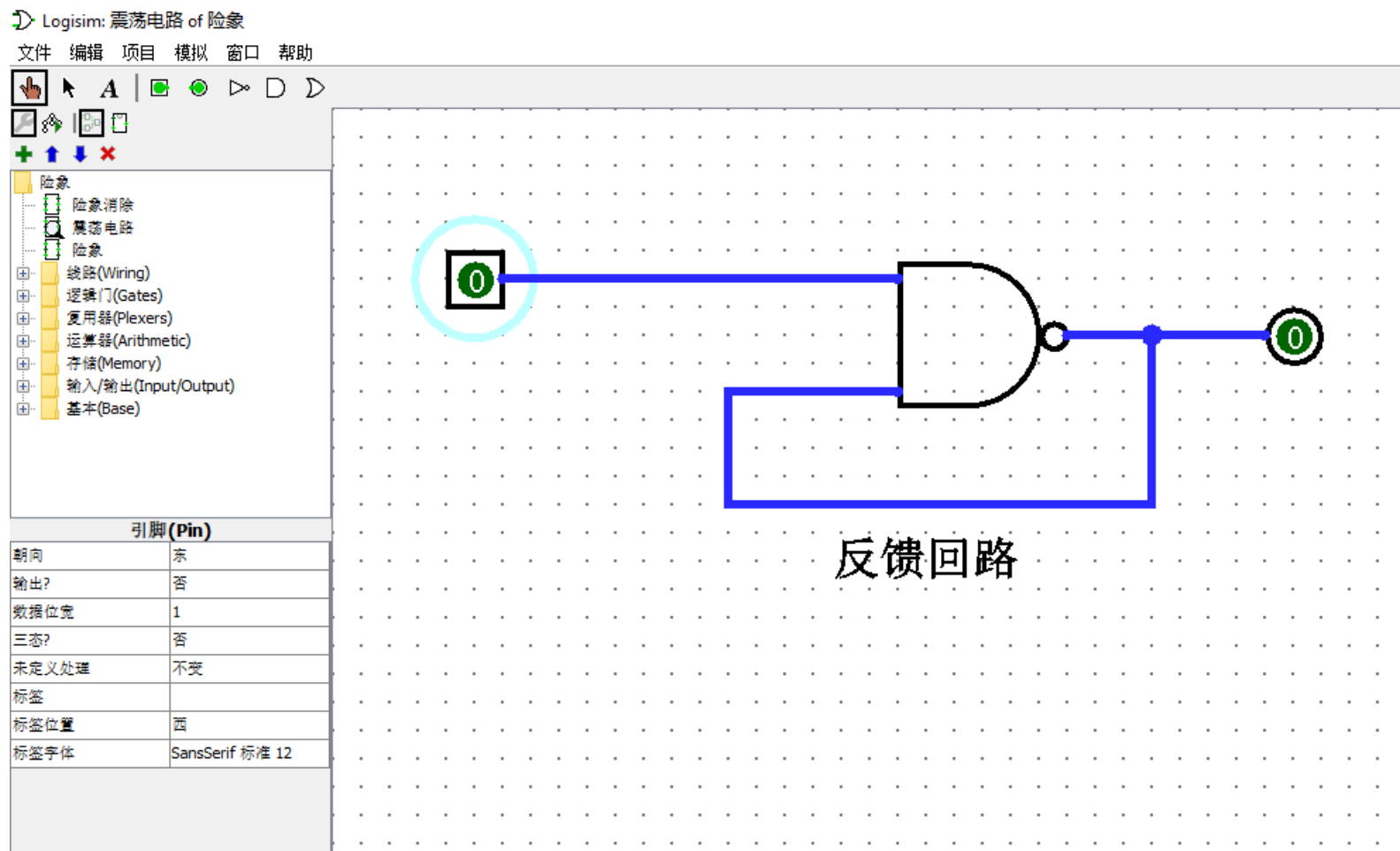


反馈回路

存在明显的震荡

■ 消除震荡

- 更改输入为0
- 重新启动持续信号模拟



Logisim: 震荡电路 of 险象

文件 编辑 项目 模拟 窗口 帮助

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)
异或非门 (XNOR Gate)

引脚 (Pin)

朝向	东
输出?	否
数据位宽	1
三态?	否
未定义处理	不变
标签	
标签位置	西
标签字体	SanSerif

133%

反馈回路



➤ Logisim常用组件库

组件通用属性



#	属性	功能描述	快捷键
1	朝向	组件在画布放置的方向	光标键
2	数据位宽	引脚对应的数据宽度	alt+ 数字
3	引脚数	逻辑门电路输入引脚数	数字键
4	外观	可以调整组件外观属性	
5	尺寸	逻辑门电路可以设置组件的尺寸大小	
6	标签	与组件相关联的标签文字，用于注释	
7	标签位置	标签在组件上的显示位置	
8	标签字体	组件标签文字的字体	



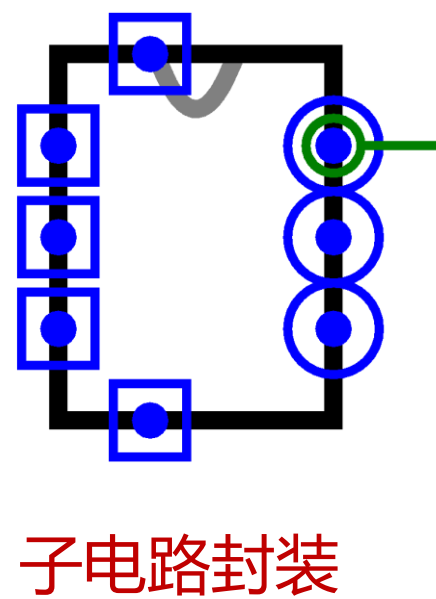
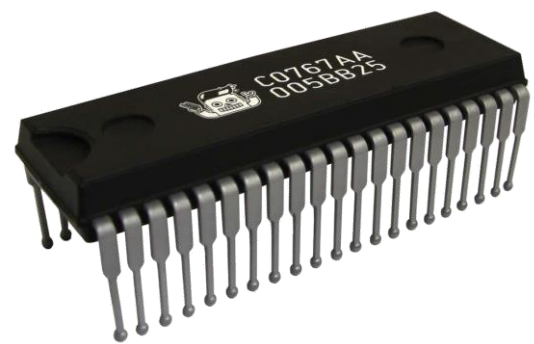
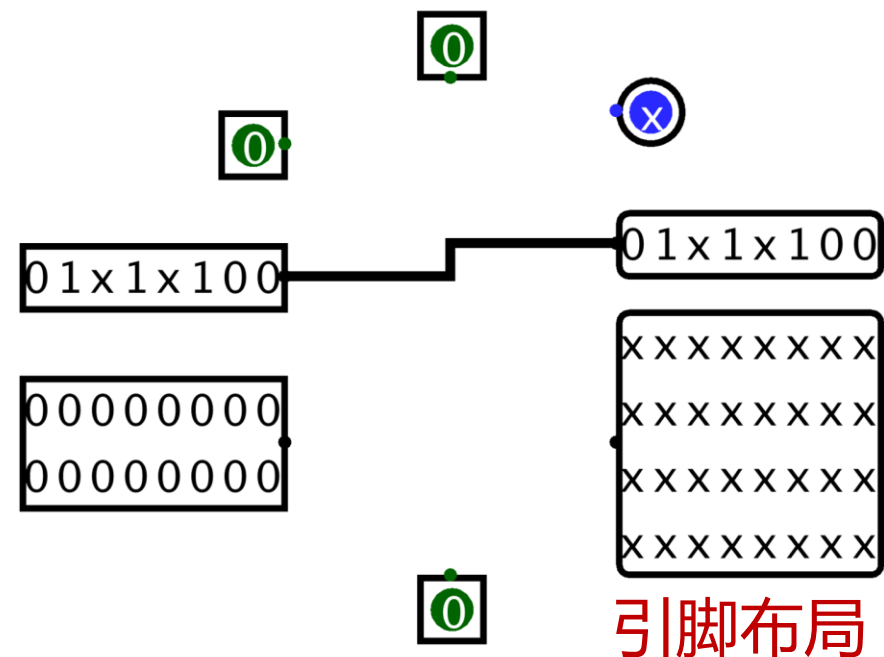
➤ Logisim常用组件库

- 线路库组件

引脚 Pin



■ 功能：子电路输入输出接口



属性	朝向	输出	数据位宽	三态
功能	器件方向	输入/输出	引脚数据位宽	输入引脚是否有三态值

探针 Probe



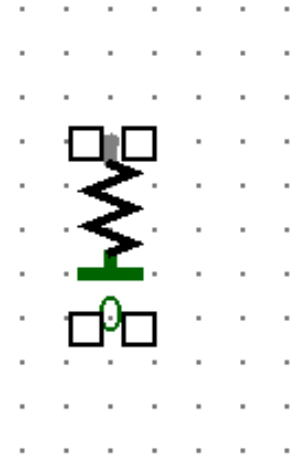
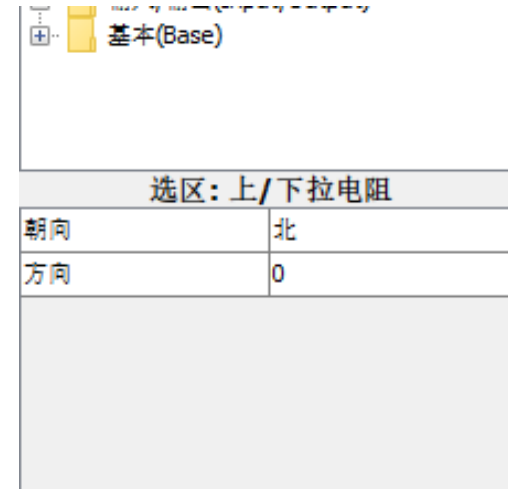
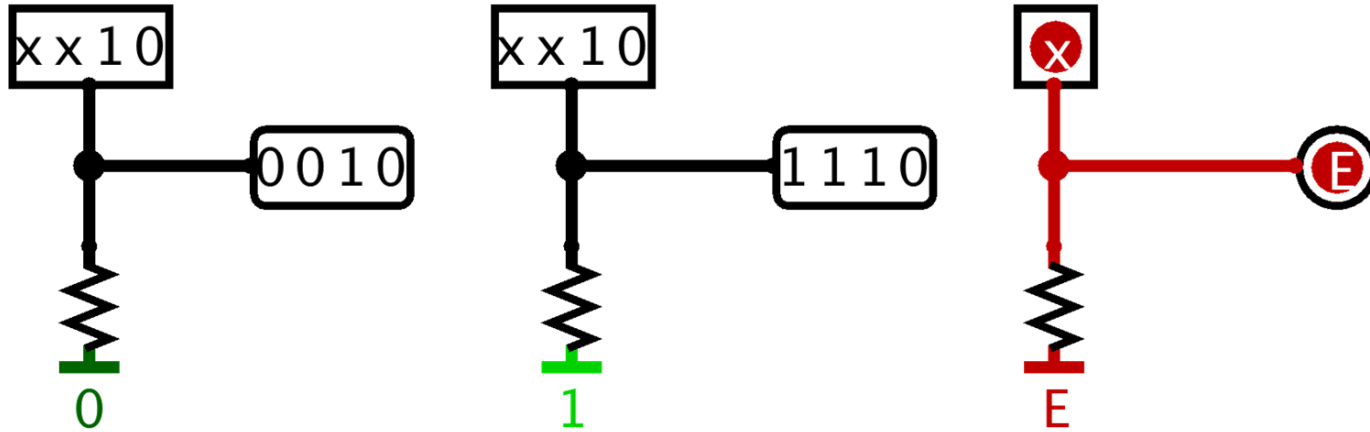
- 功能：动态监测线路值



上/下拉电阻 Pull Resistor



- 功能：处理线路中的悬浮态、不确定值
 - 无位宽属性
 - 更改上拉/下拉，可通过更改属性中的方向实现



常量、接地、电源



■ 常量



■ 电源、接地





➤ Logisim常用组件库

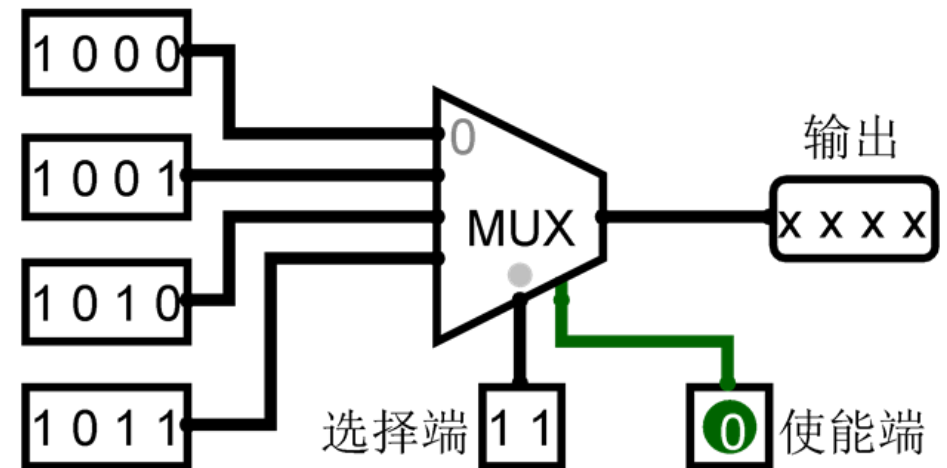
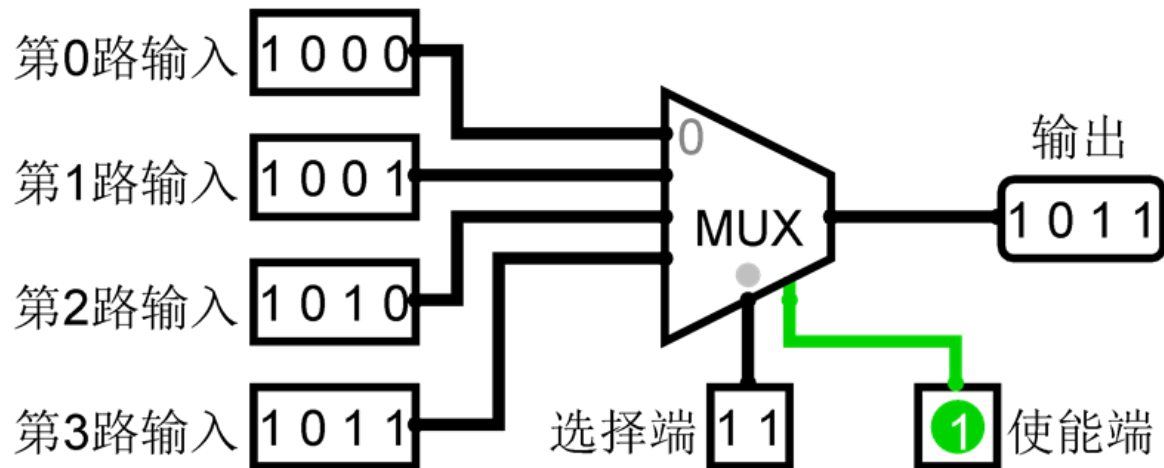
- 复用器库

多路选择器MUX



■ 功能：从多路输入中选择一路（通过选择端选择）进行输出

- 选择端位宽为 n ，则输入源的数目为 2^n
- 使能端默认为1，可关闭，避免误接



不恰当的级联

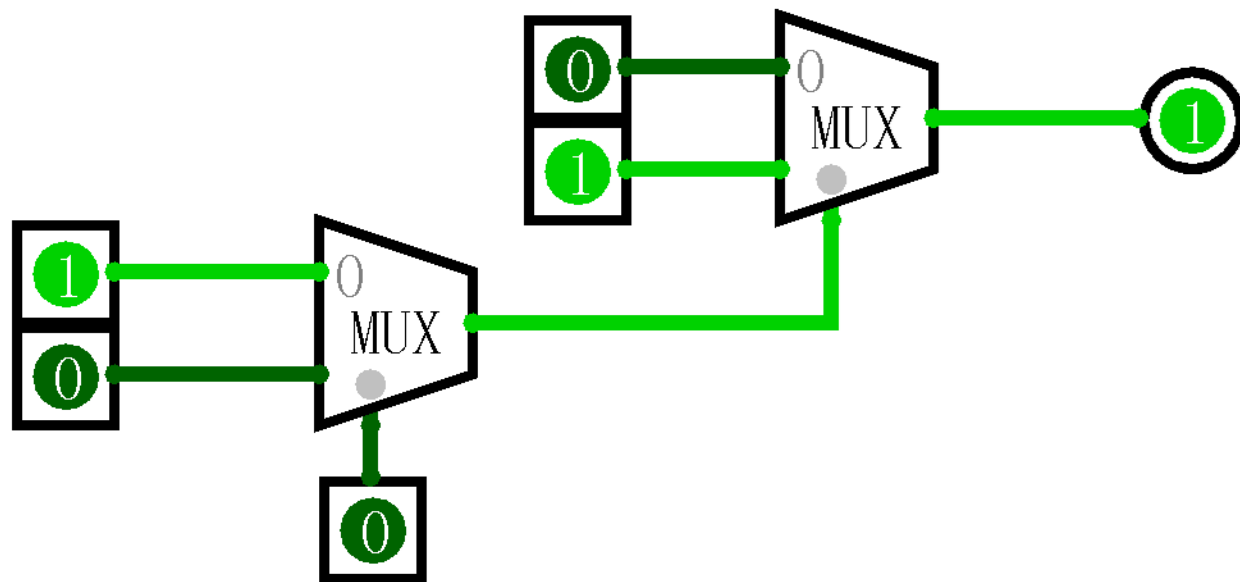


■ 不要将MUX当if-else的实现组件使用

- 用多路选择器代替逻辑电路 (x)
- MUX输出级联另一MUX选择端 (x)
- 多路选择器本身有时间延迟，采用级联会增大整个电路的延迟

■ 用组合逻辑的基本门电路来完成

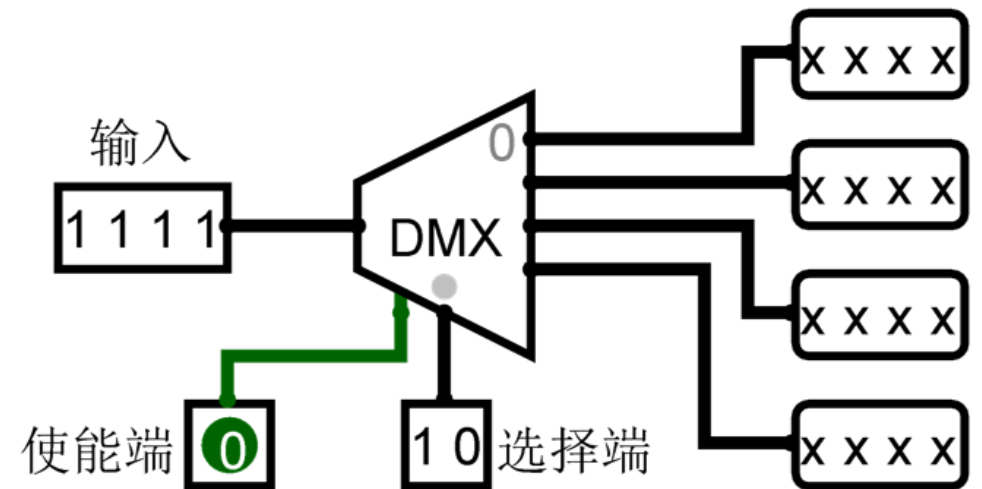
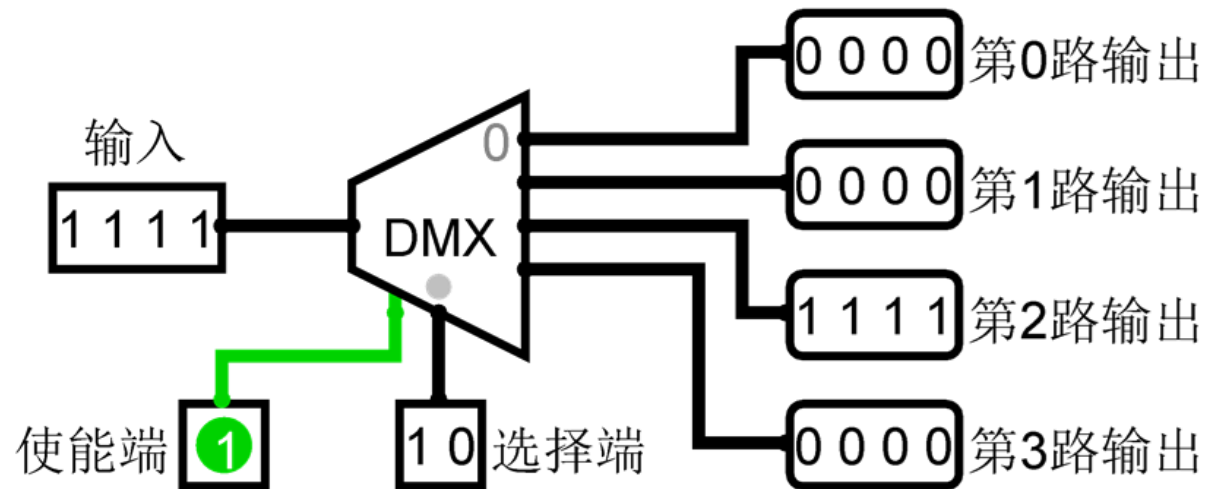
- 数字逻辑的标准设计方法



解复用器DMX



- 功能：将一路输入输出到多路输出中的其中一路（通过选择端选择）





谢谢！