《计算机组成原理实验》

（预备实验）

厦门大学信息学院软件工程系　曾文华

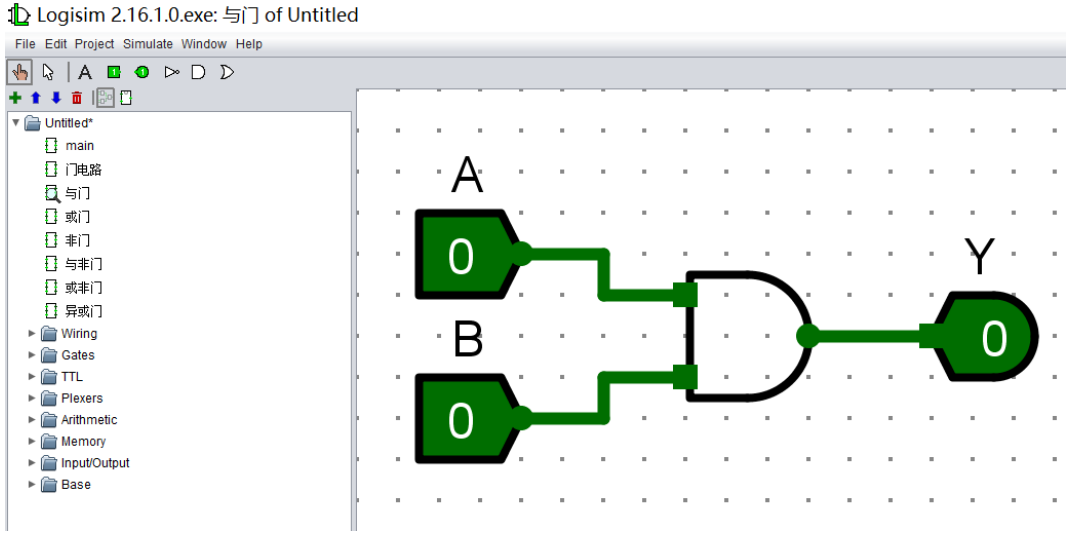
2023年2月27日

预备实验　Logisim的使用

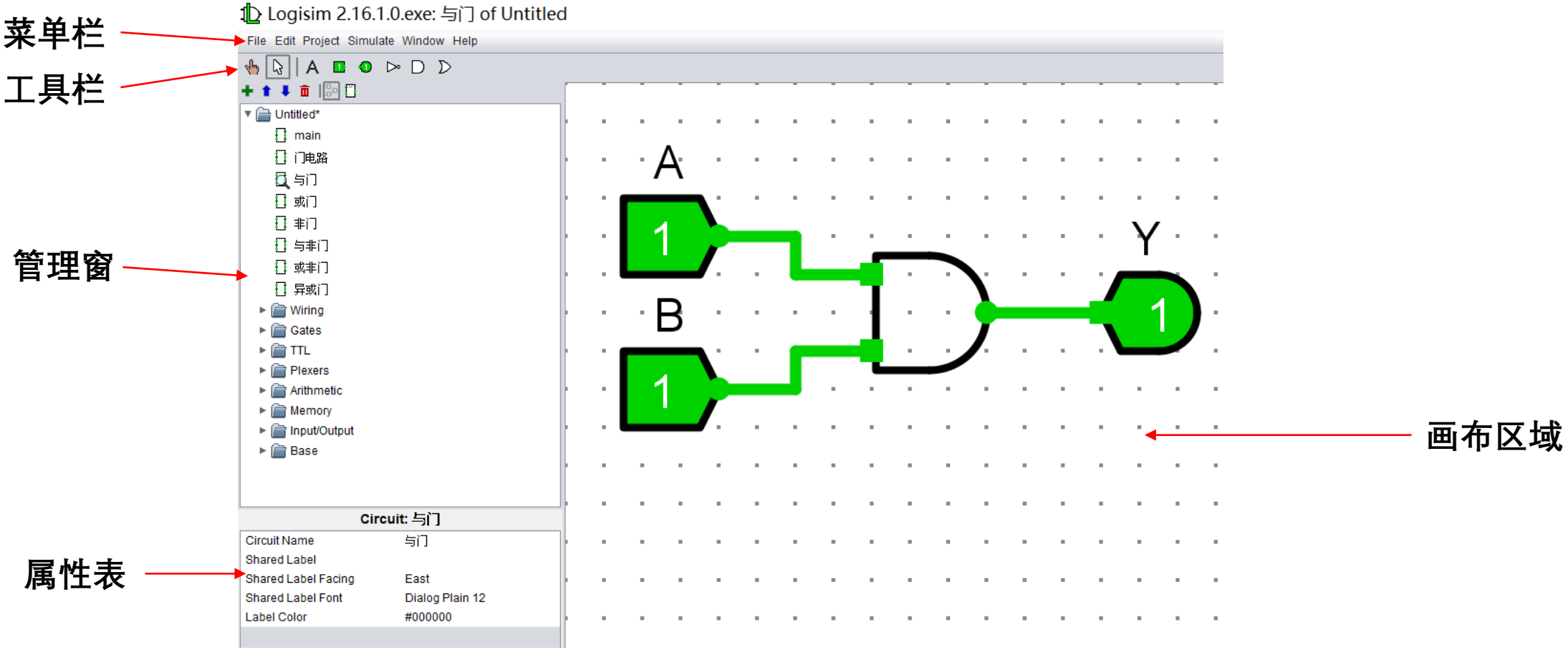
参考设计文件：Logisim的使用.circ（请从FTP下载）

Logisim简介

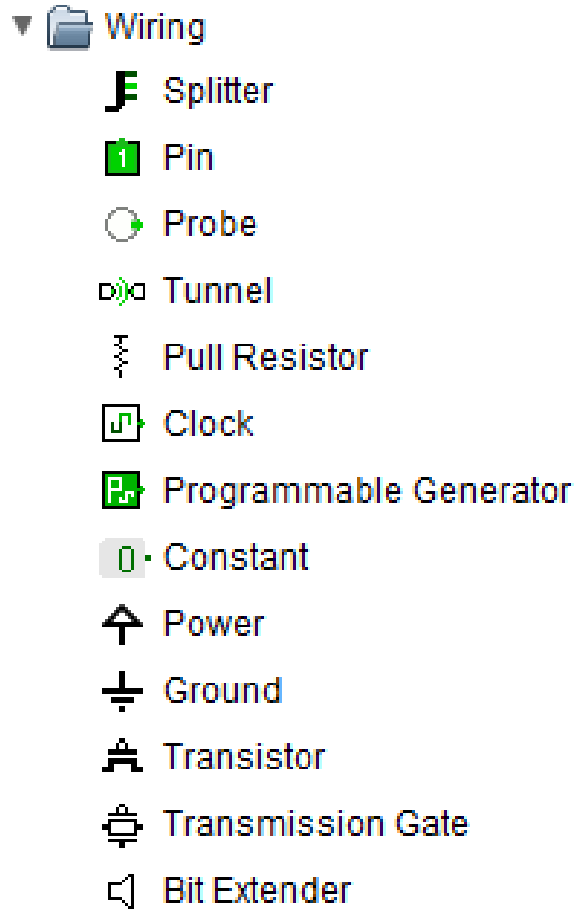
- Logisim是一种用于设计和模拟数字逻辑电路的教育工具。凭借其简单的工具栏界面和构建它们时的电路仿真，它非常简单，有助于学习与逻辑电路相关的最基本概念。由于能够从较小的子电路构建更大的电路，并通过鼠标拖动来绘制电线束，因此可以使用Logisim来设计和仿真用于相关课程的教学目的。



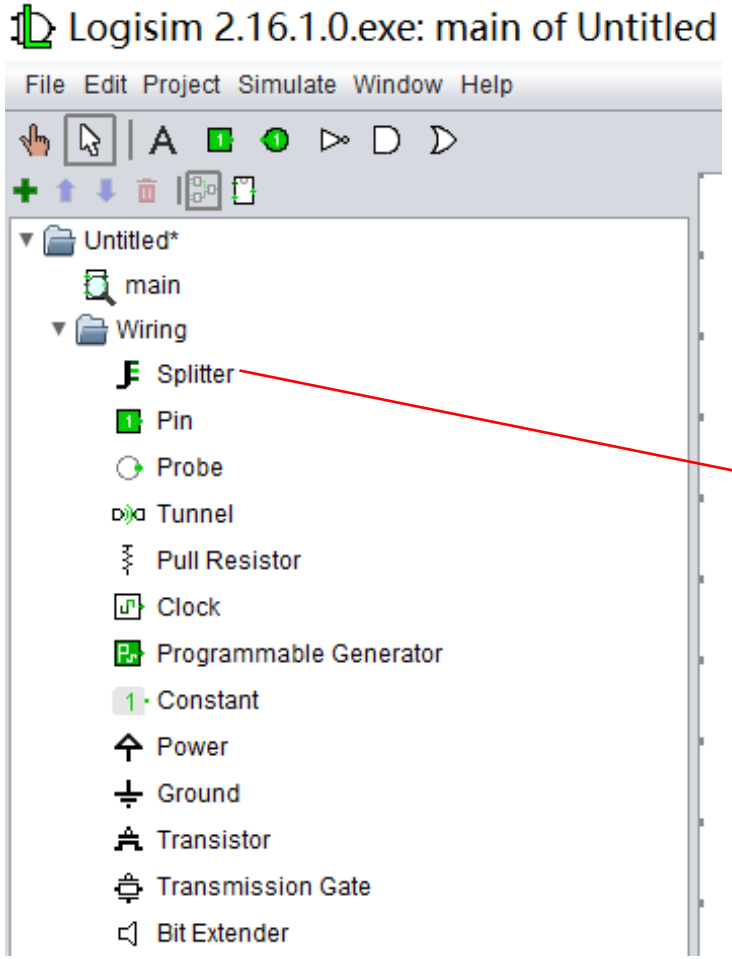
- 从FTP下载“logisim-hust-20200118.exe”，点击直接运行。

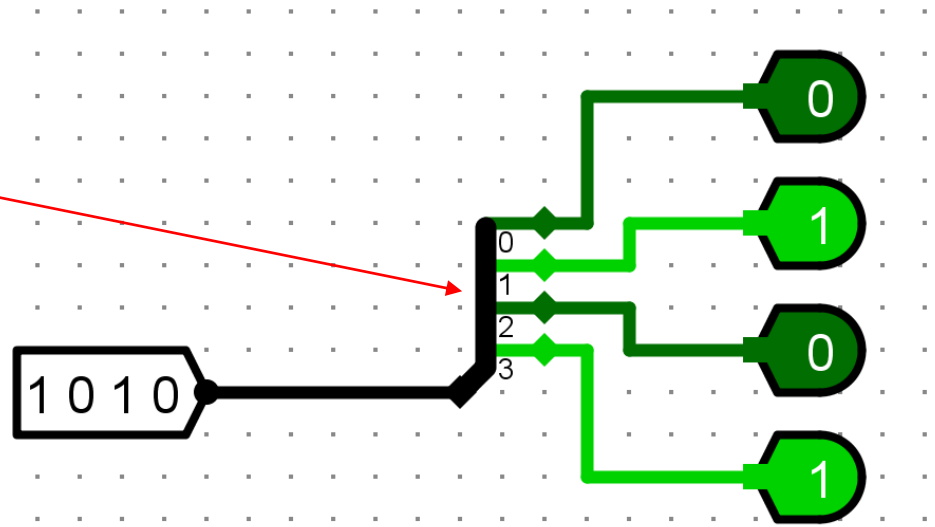


（一）线路（Wiring）


Wiring
Splitter
Pin
Probe
Tunnel
Pull Resistor
Clock
Programmable Generator
Constant
Power
Ground
Transistor
Transmission Gate
Bit Extender

分线器（Splitter）——用于输出



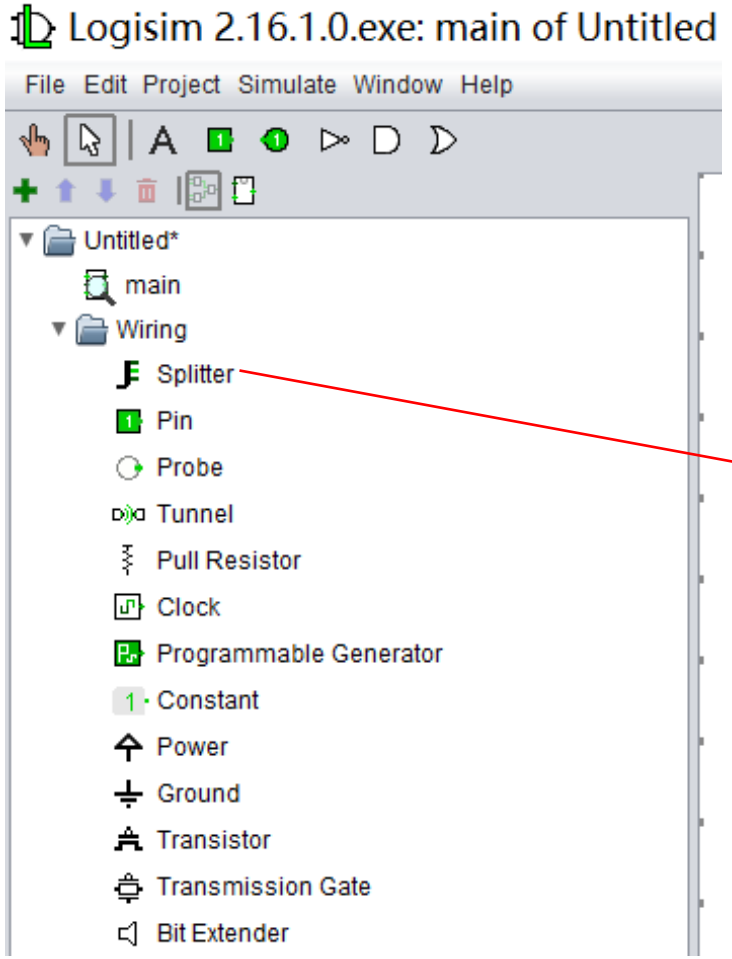


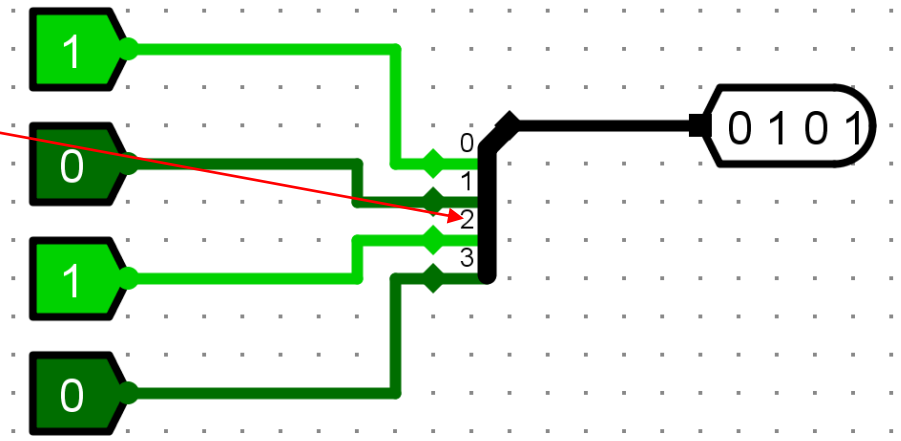
一变多

Splitter的属性（可以改变）

Selection: Splitter	
Facing	East
Fan Out	4
Bit Width In	4
Appearance	Left-handed
Bit 0	0 (Top)
Bit 1	1
Bit 2	2
Bit 3	3 (Bottom)

分线器（Splitter）——用于输入



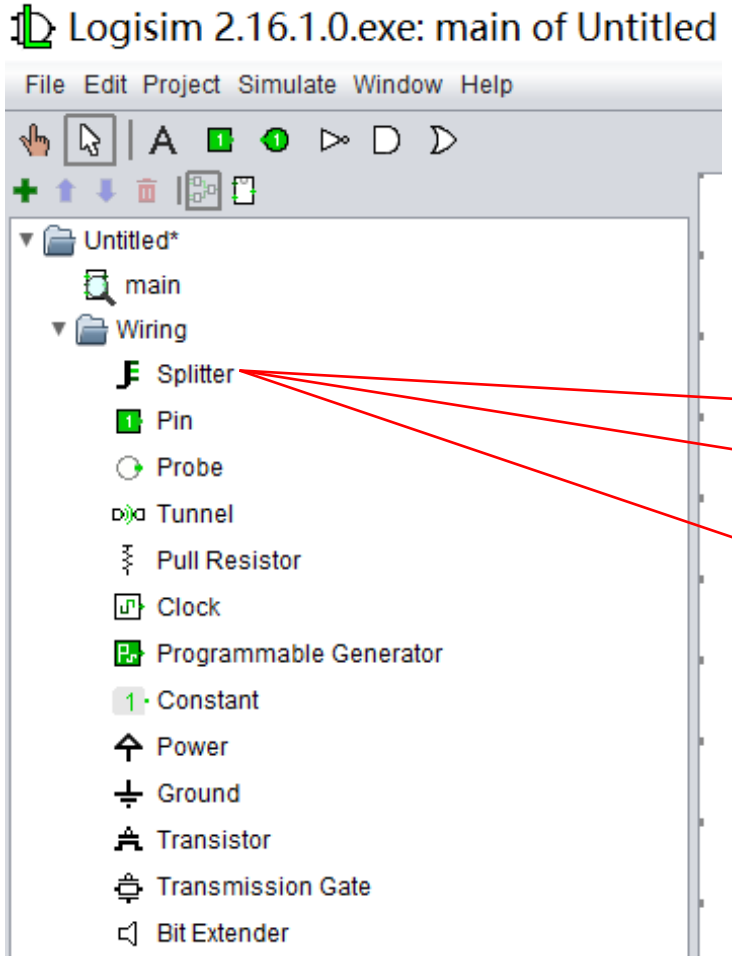


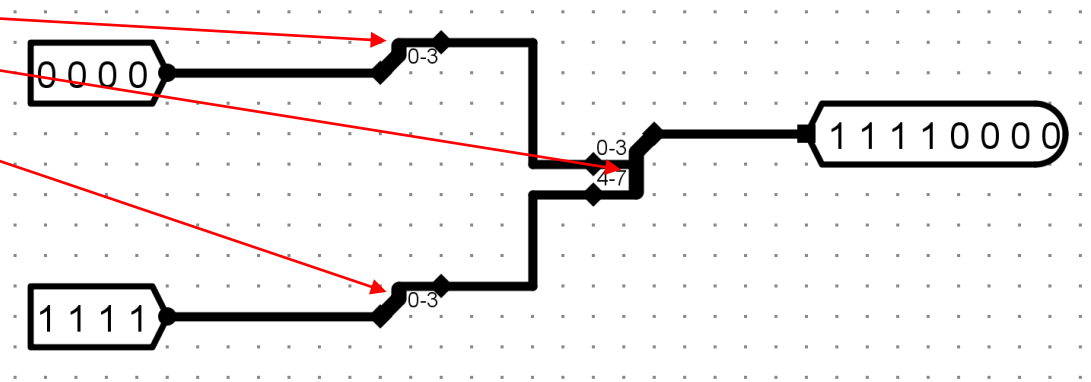
多变一

Splitter的属性（可以改变）

Selection: Splitter	
Facing	East
Fan Out	4
Bit Width In	4
Appearance	Left-handed
Bit 0	0 (Top)
Bit 1	1
Bit 2	2
Bit 3	3 (Bottom)

分线（Splitter）——拼接



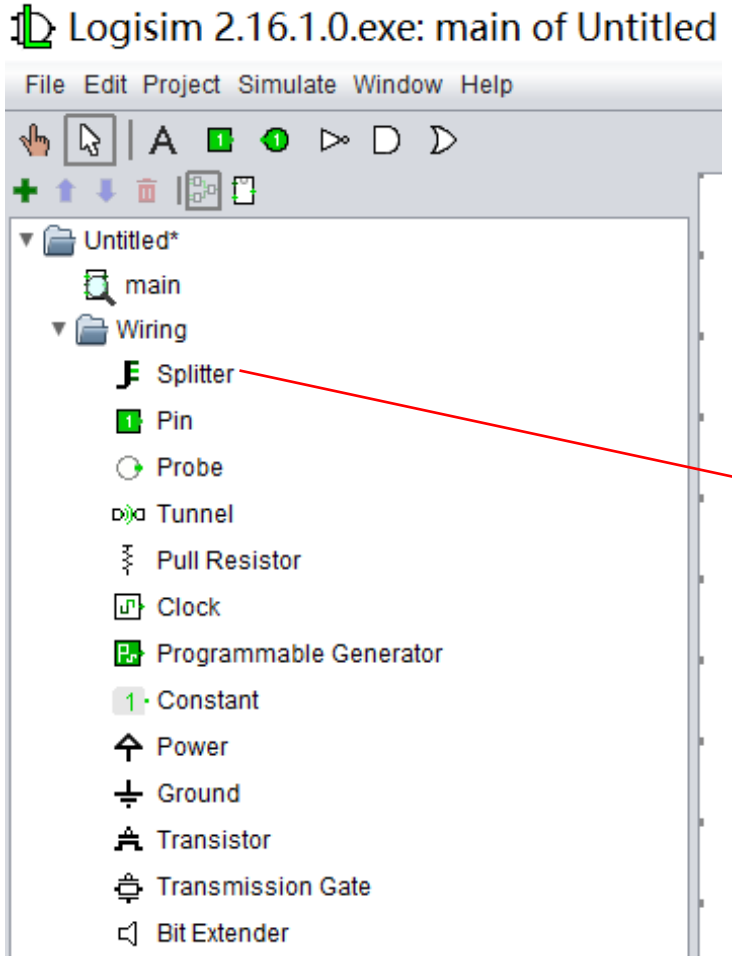


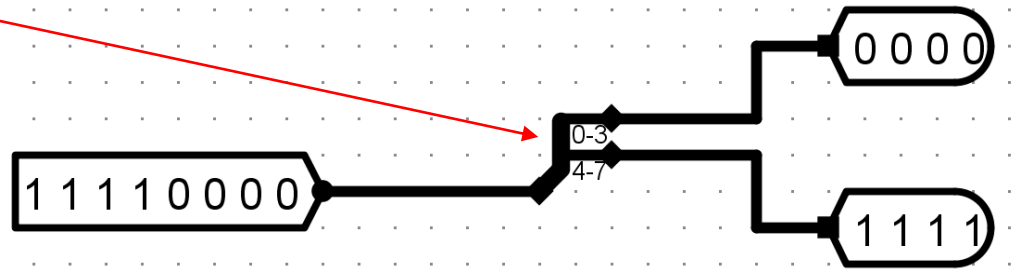
2个4位输入拼接成1个8位输出

Splitter的属性（可以改变）

Selection: Splitter	
Facing	East
Fan Out	4
Bit Width In	4
Appearance	Left-handed
Bit 0	0 (Top)
Bit 1	1
Bit 2	2
Bit 3	3 (Bottom)

分线器（Splitter）——分线





1个8位输入分成2个4位输出

Splitter的属性（可以改变）

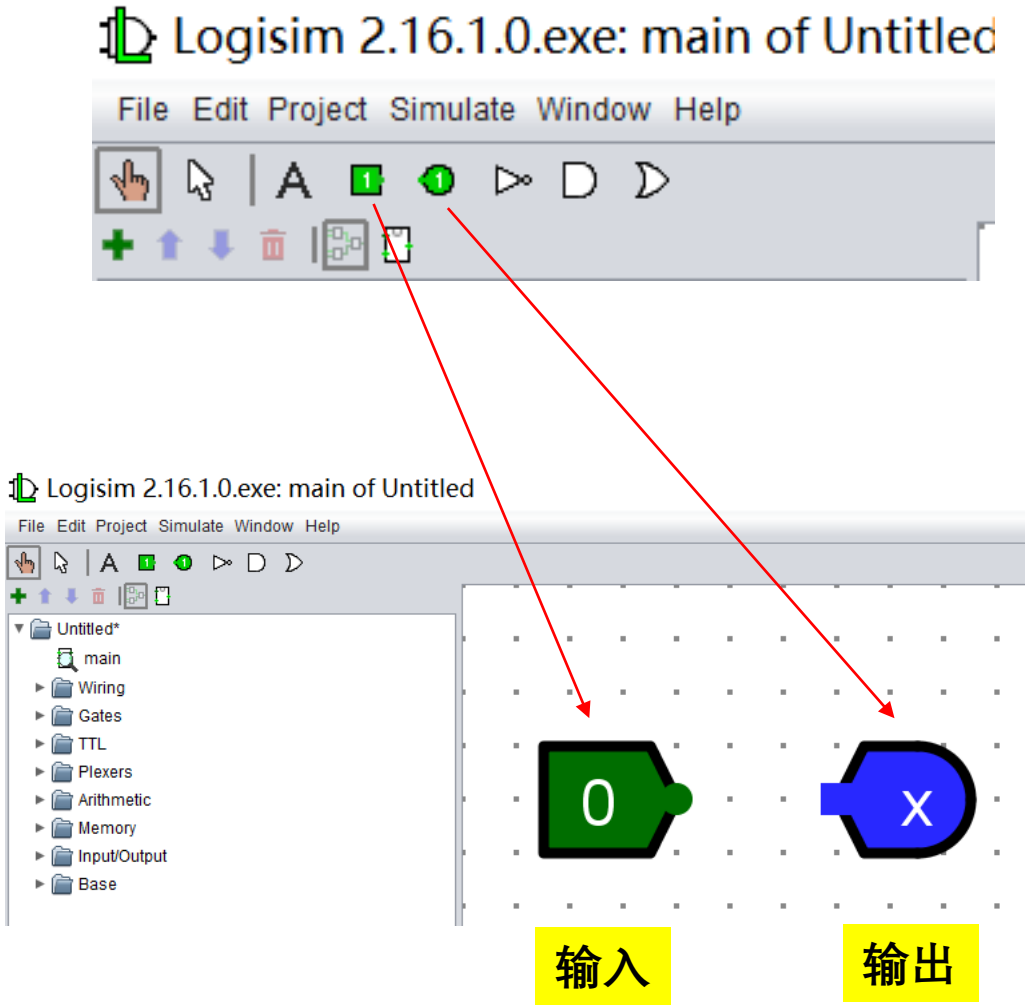
Selection: Splitter	
Facing	East
Fan Out	4
Bit Width In	4
Appearance	Left-handed
Bit 0	0 (Top)
Bit 1	1
Bit 2	2
Bit 3	3 (Bottom)

输入、输出（Pin）

Pin的属性（可以改变）

Selection: Pin	
Facing	East
Output?	No
Data Bits	1
Three-state?	No
Pull Behavior	Unchanged
Label	
Label Location	North
Label Font	Dialog Plain 12
Label Color	#000000

输入

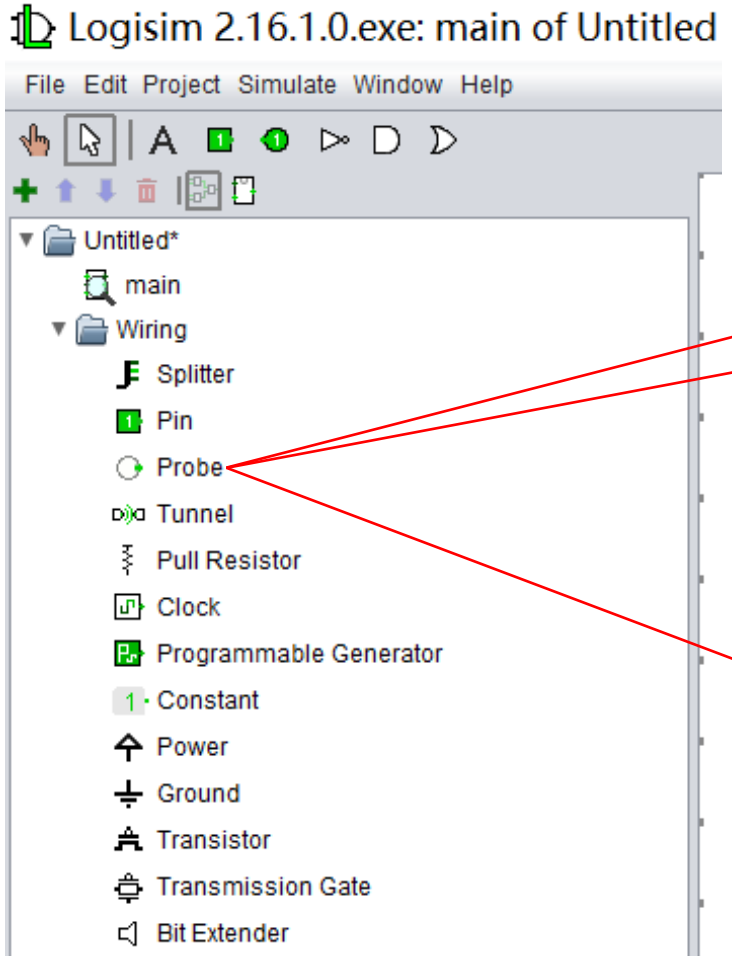


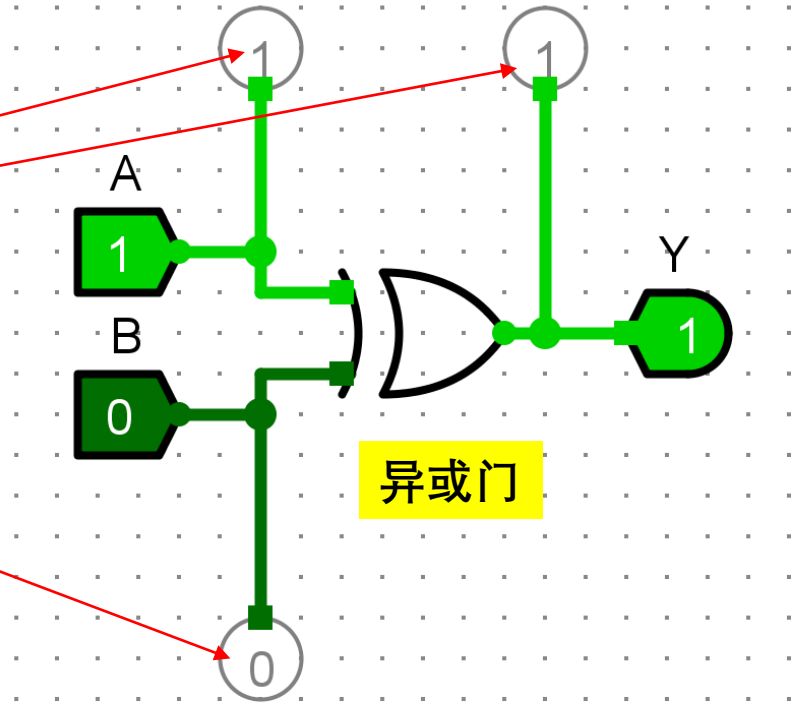
Pin的属性（可以改变）

Pin	
Facing	West
Output?	Yes
Data Bits	1
Three-state?	Yes
Pull Behavior	Unchanged
Label	
Label Location	North
Label Font	Dialog Plain 12
Label Color	#000000

输出

探针（Probe）

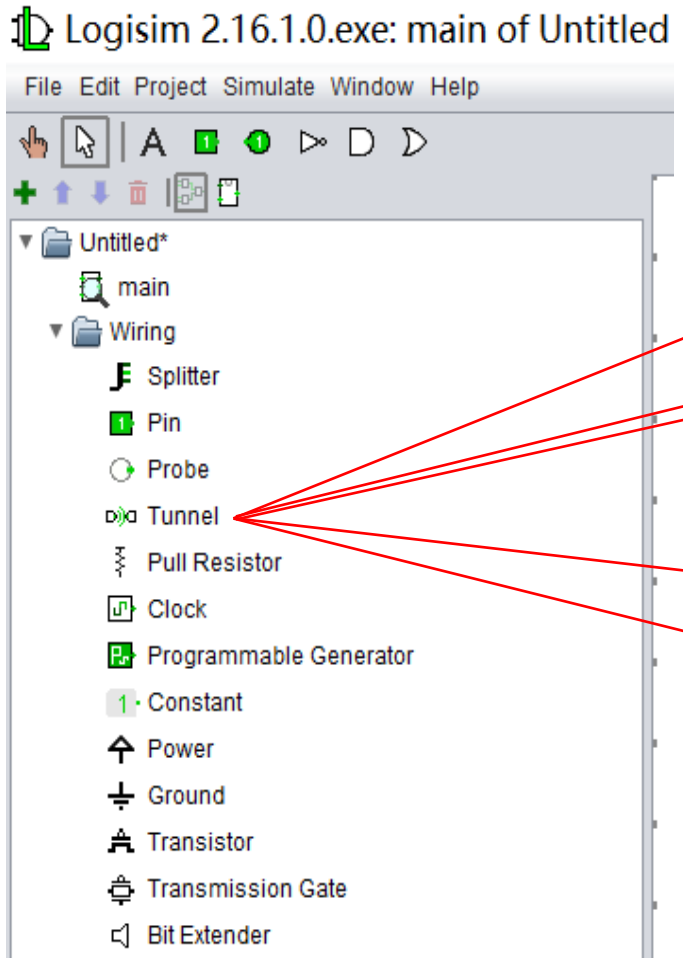


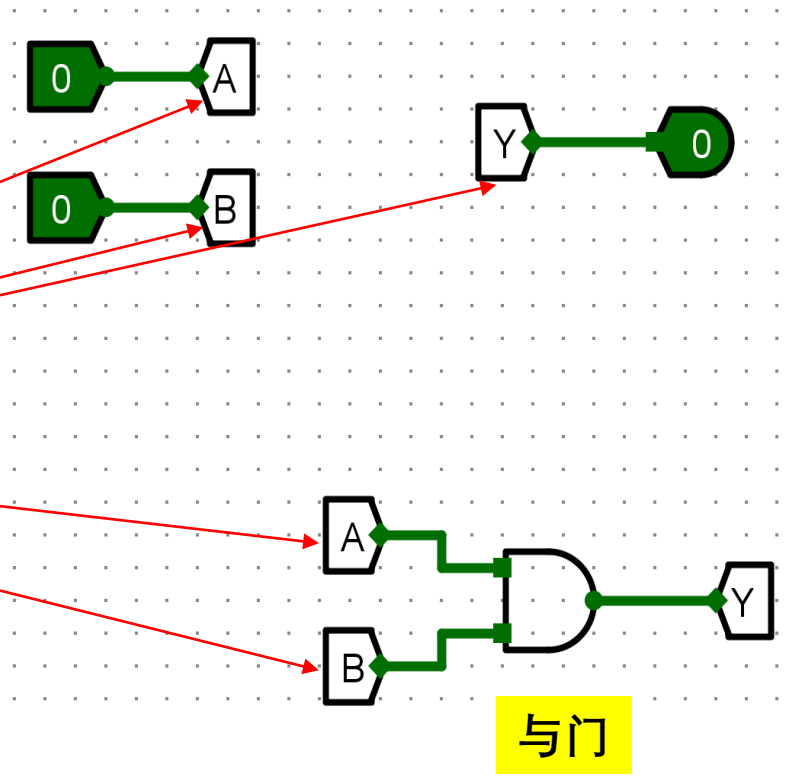


Probe的属性（可以改变）

Selection: Probe	
Facing	South
Radix	Unsigned Decimal
Label	
Label Location	North
Label Font	Dialog Plain 12
Label Color	#000000

隧道（Tunnel）

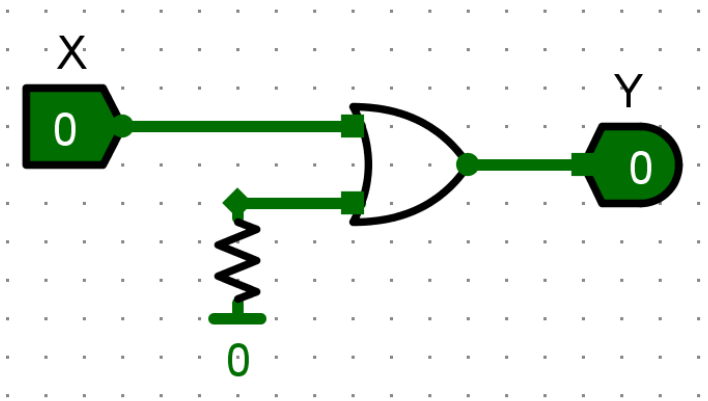




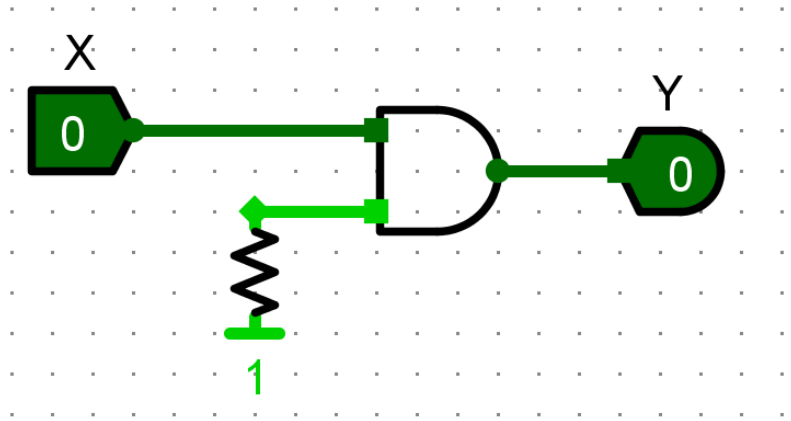
Tunnel的属性（可以改变）

Selection: Tunnel	
Facing	West
Data Bits	1
Label	B
Label Font	Dialog Plain 12

上拉电阻（Pull Resistor）

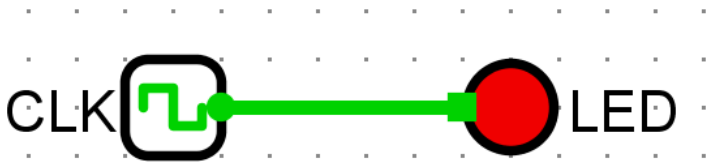


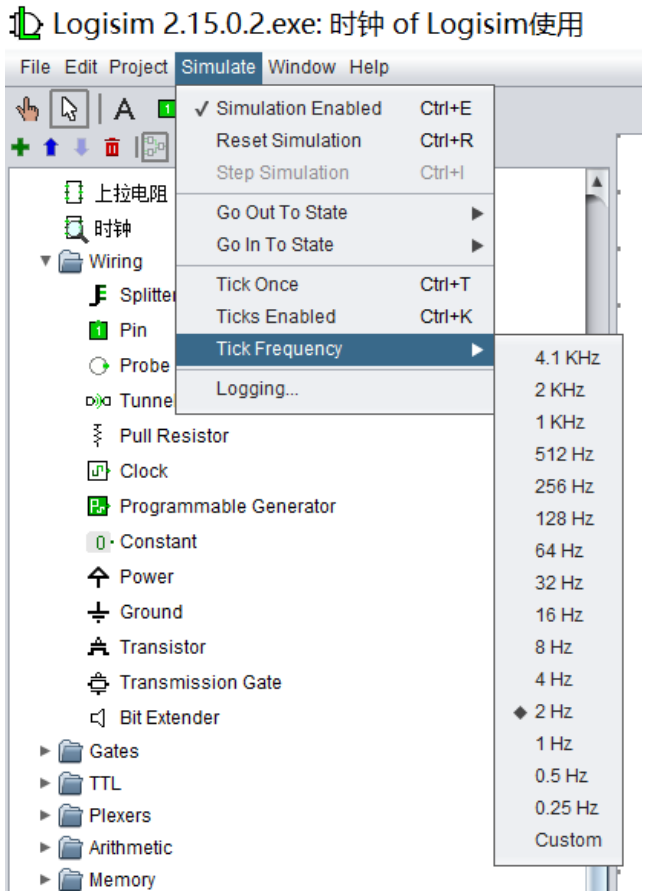
Y = X+0 = X



Y = X · 1 = X

时钟（Clock）





按Ctrl+T：产生1次时钟

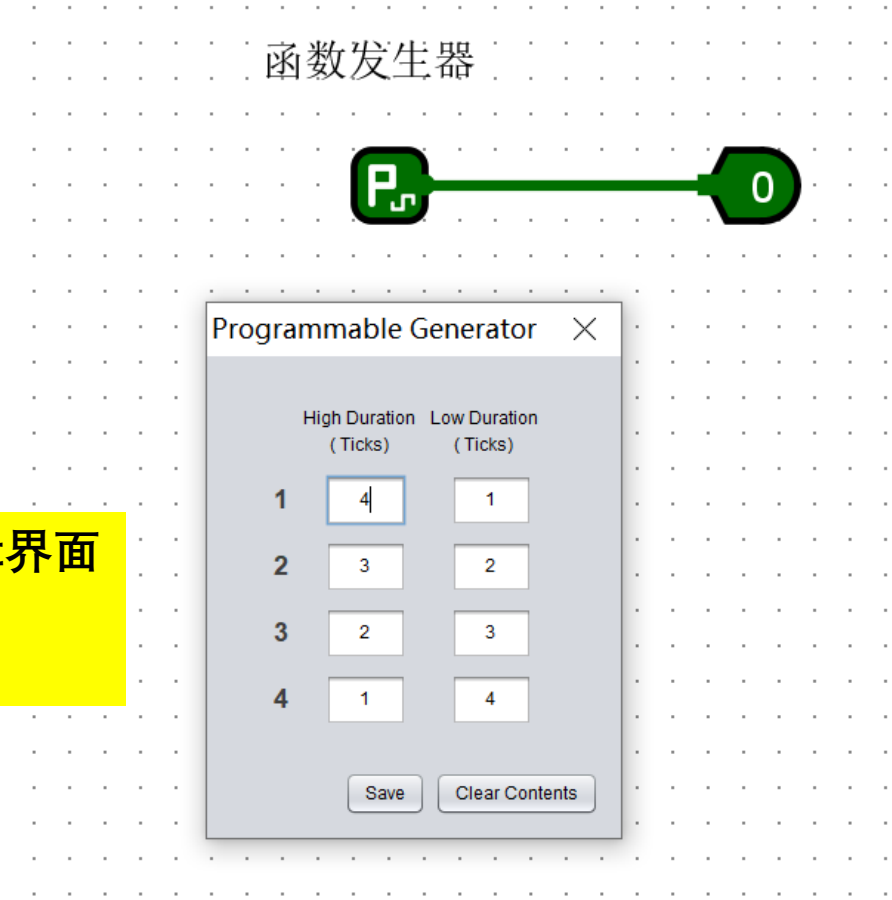
按Ctrl+K：产生连续的时钟

时钟频率可以设置

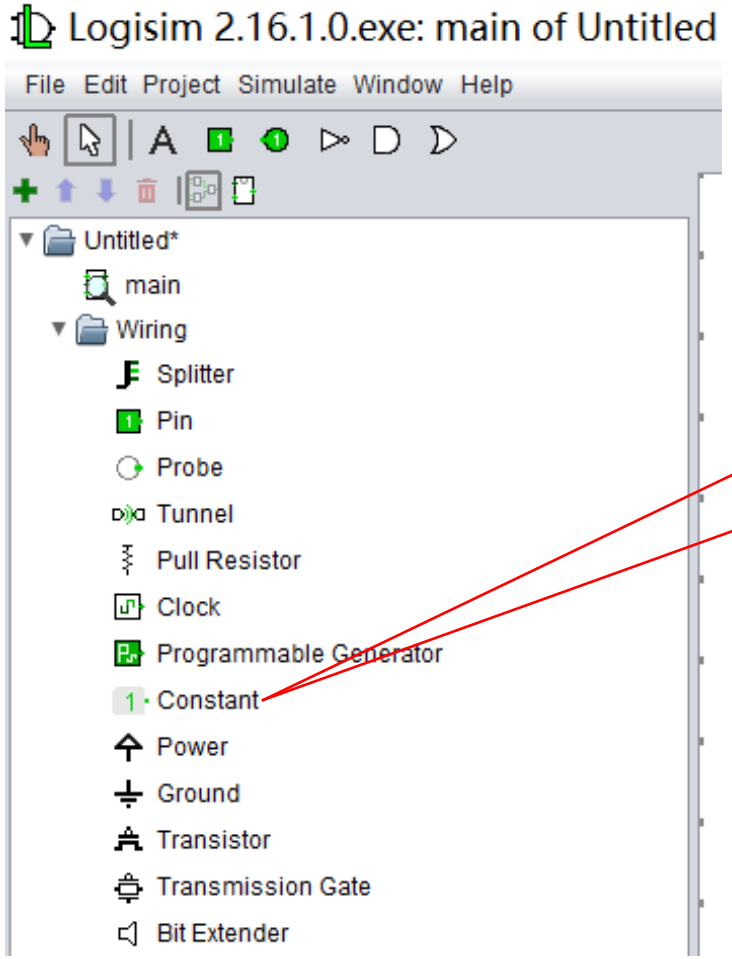
函数发生器（Programmable Generator）

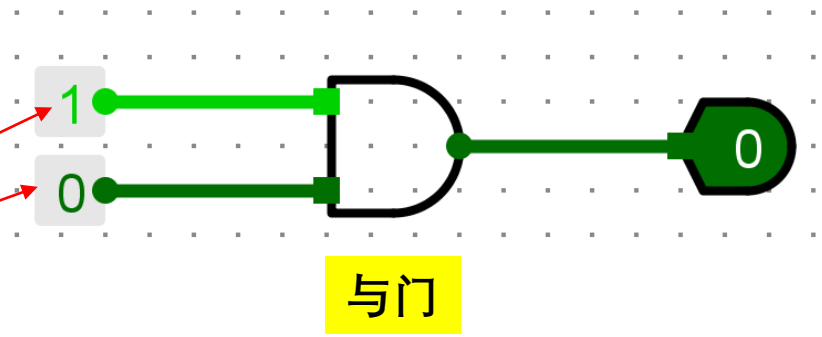
点击右键得到这个编辑界面

可以修改其中的内容



常量（Constant）

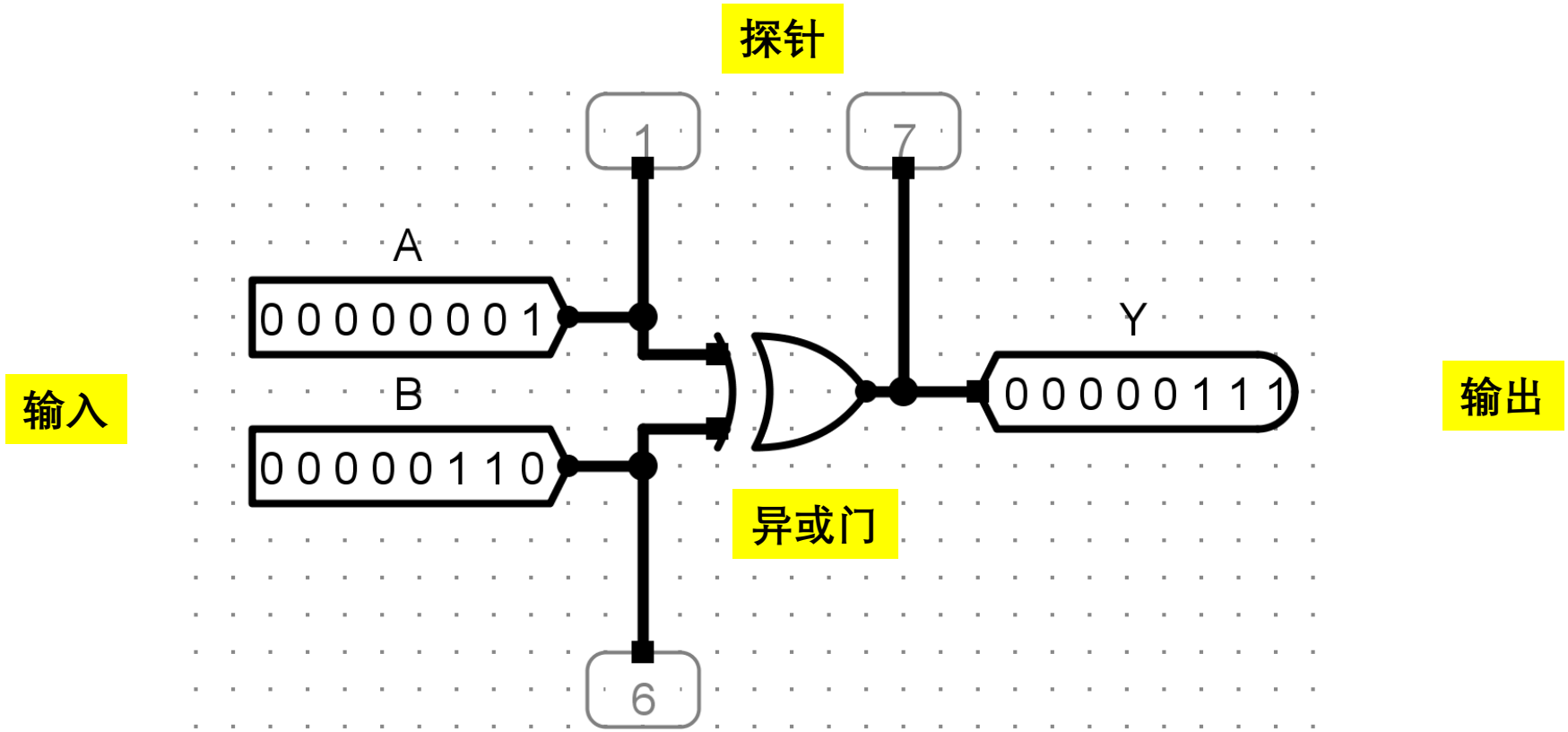




Constant的属性（可以改变）

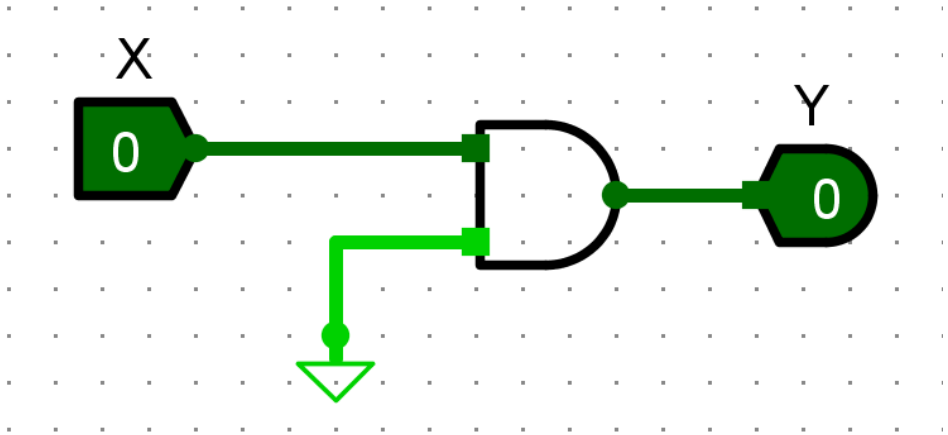
Selection: Constant	
Facing	East
Data Bits	1
Value	0x1

常量（Constant）

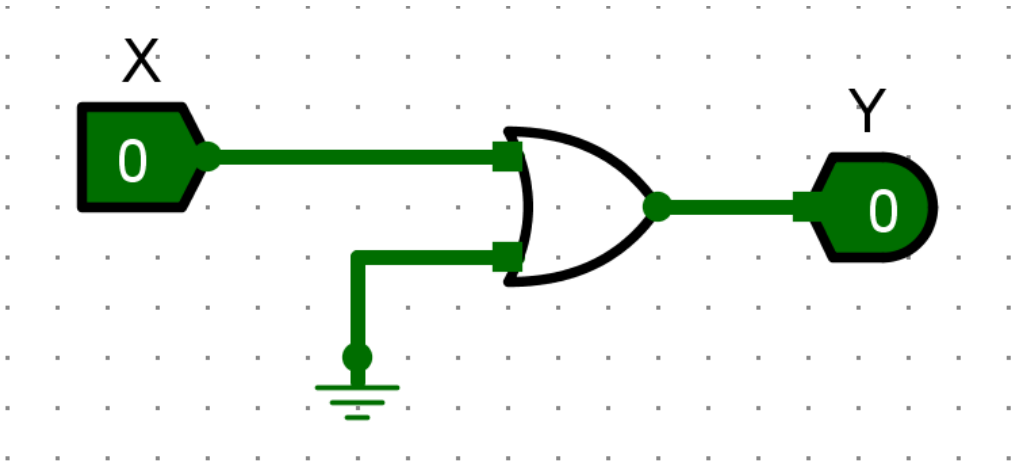


分别改变2个输入、1个输出、异或门的属性，使其位宽=8

电源（Power）、接地（Ground）

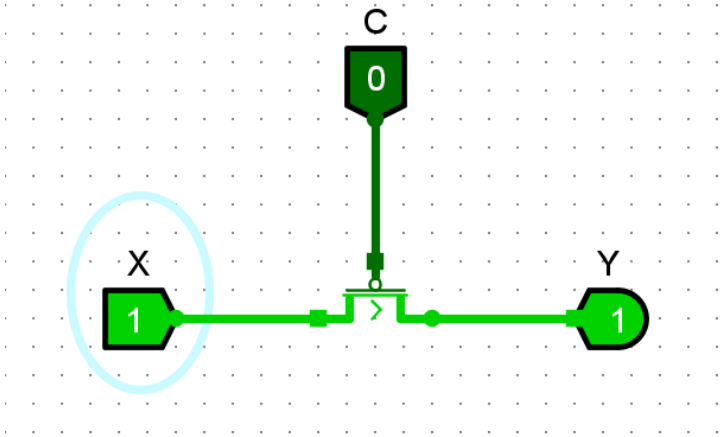


$Y = X \cdot 1 = X$

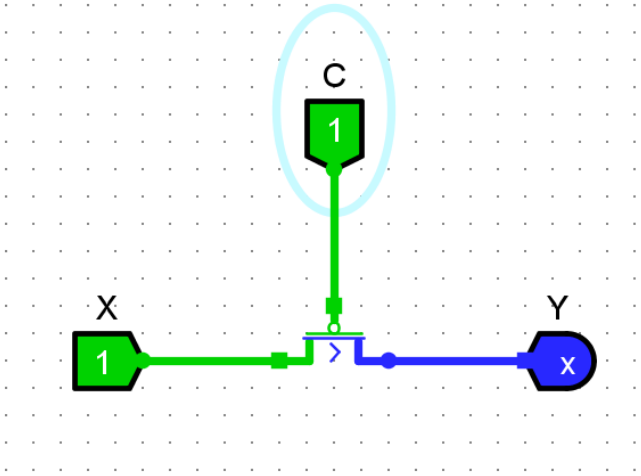


$Y = X+0 = X$

晶体管（Transistor）

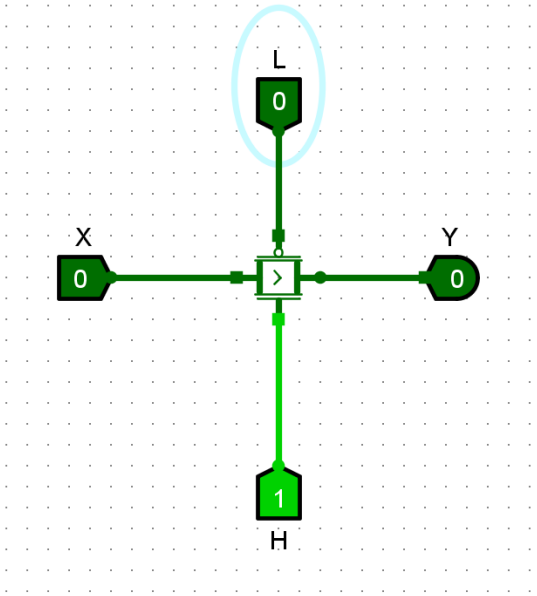


C=0　Y=X

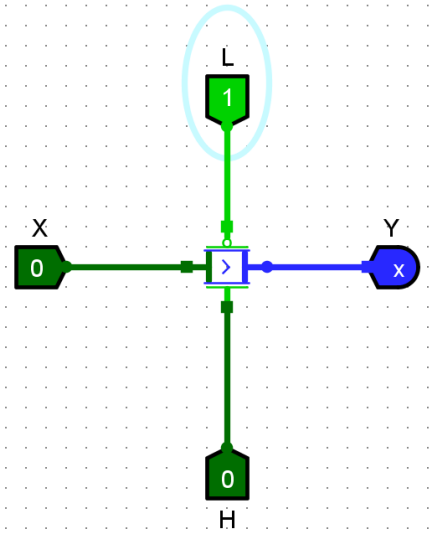


C=1　Y=高阻

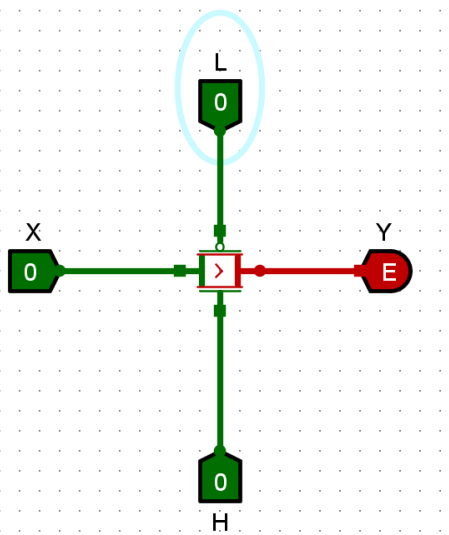
传输门（Transmission Gate）



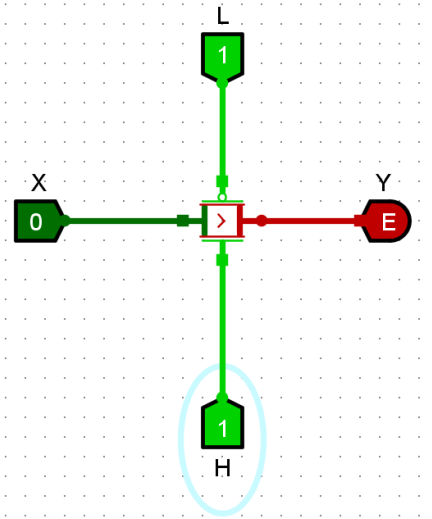
H=1、L=0　Y=X



H=0、L=1　Y=高阻

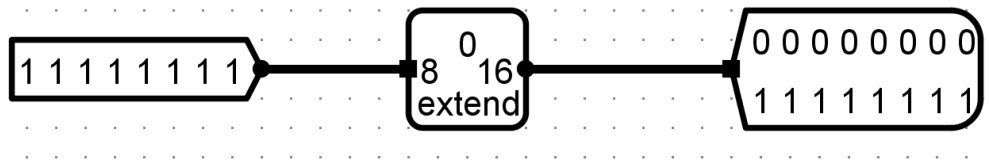


H=0、L=0　Y出错

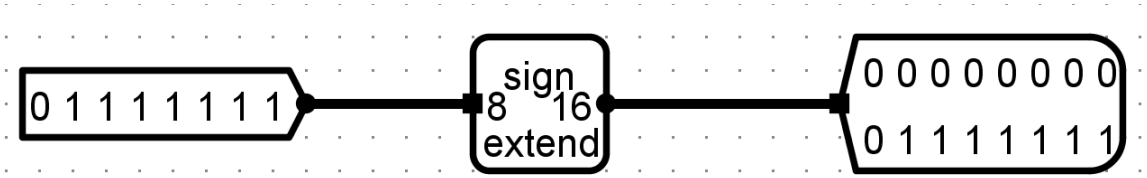


H=1、L=1　Y出错

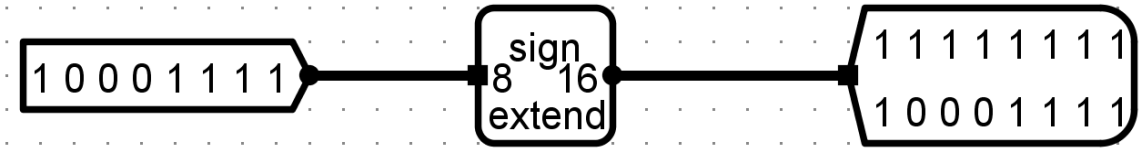
位扩展器（Bit Extender）



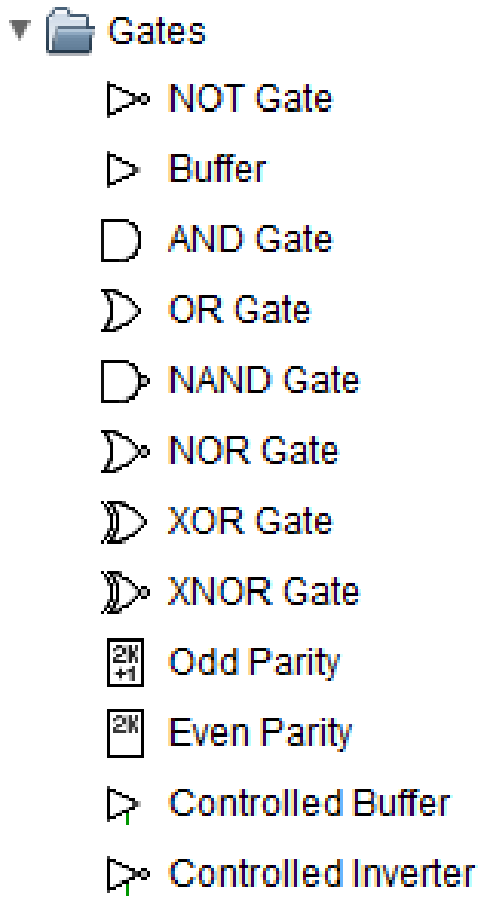
零扩展



符号扩展

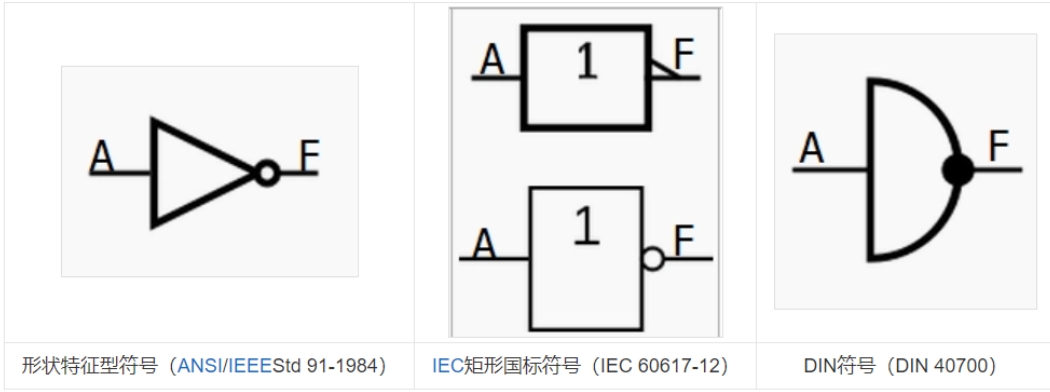


（二）门电路（Gates）


Gates
NOT Gate
Buffer
AND Gate
OR Gate
NAND Gate
NOR Gate
XOR Gate
XNOR Gate
Odd Parity
Even Parity
Controlled Buffer
Controlled Inverter

非门（NOT Gate）

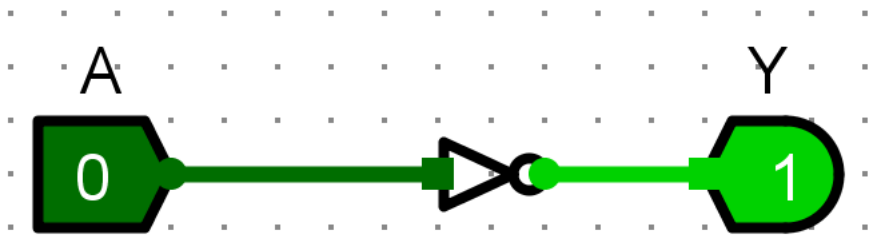
- 非门：$Y=\overline{A}$



形状特征型符号（ANSI/IEEEStd 91-1984）　IEC矩形国标符号（IEC 60617-12）　DIN符号（DIN 40700）

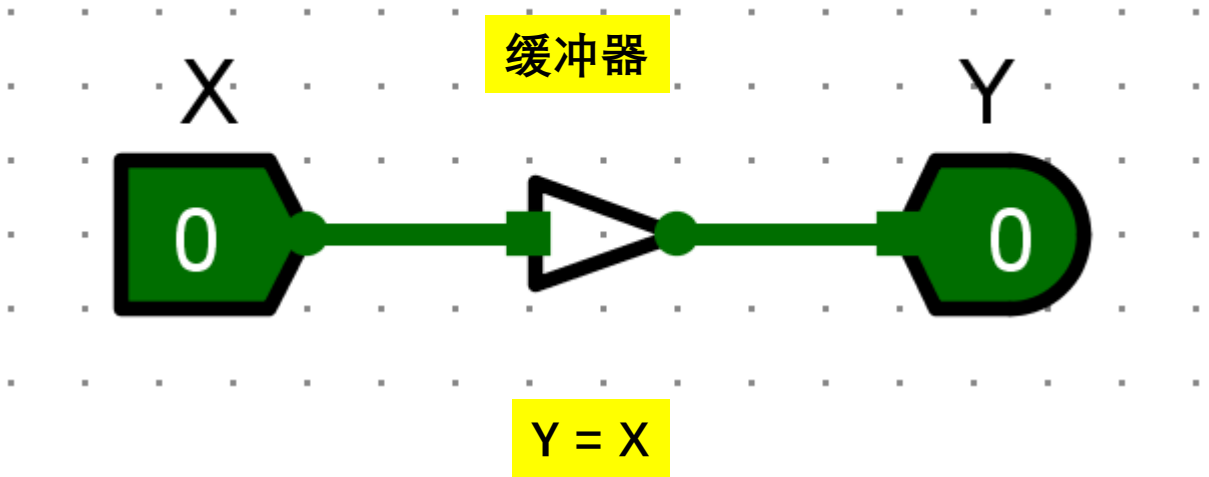
真值表

输入A	输出Y
0	1
1	0



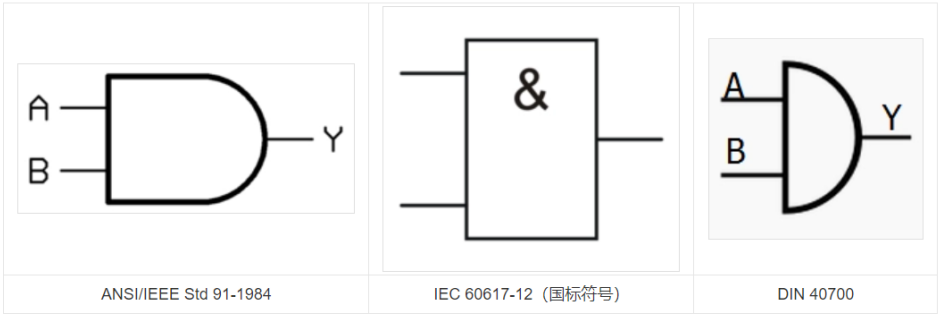
非门

缓冲器（Buffer）



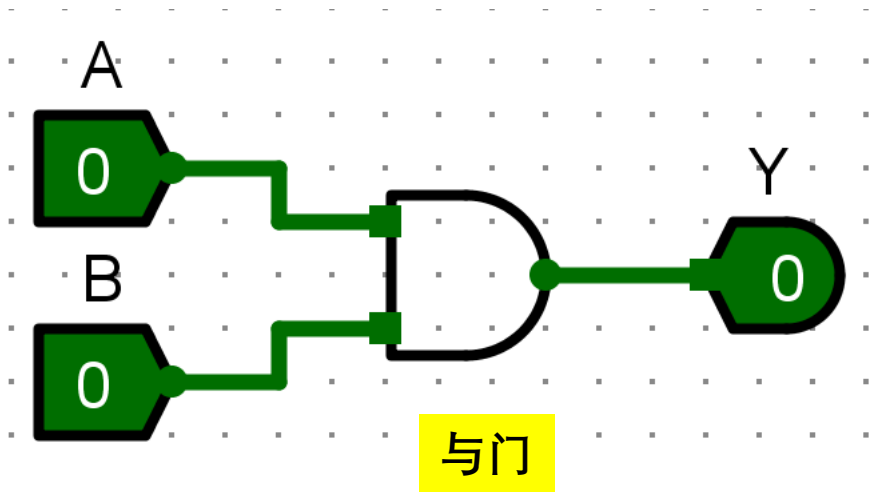
与门（AND Gate）

- 与门：Y=A·B



真值表

输入A	输入B	输出Y
0	0	0
0	1	0
1	0	0
1	1	1



与门

AND Gate	
Facing	East
Data Bits	1
Gate Size	Narrow
Number Of Inputs	2
Output Value	0/1
Label	
Label Font	Dialog Plain 12
Label Color	#000000
Negate 1 (Top)	No
Negate 2 (Bottom)	No

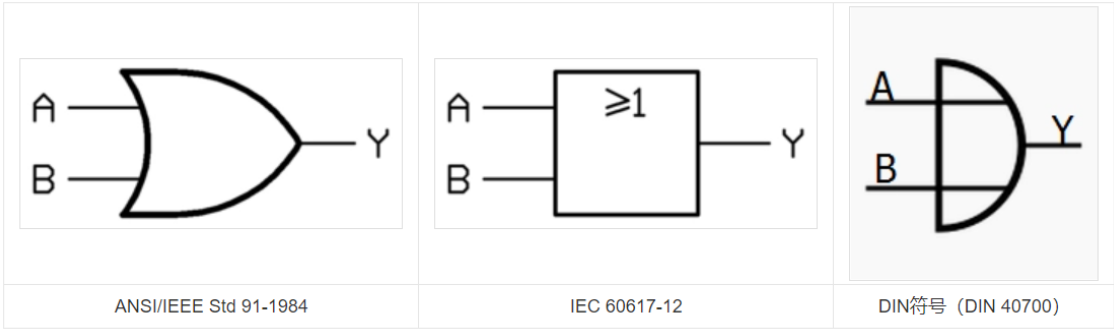
与门的属性

或门（OR Gate）

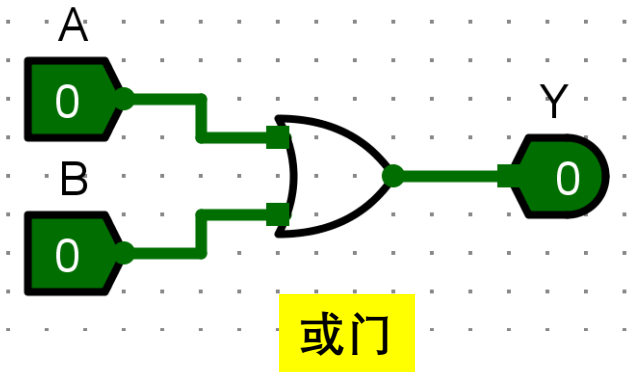
- 或门：Y=A+B

真值表

输入A	输入B	输出F
0	0	0
0	1	1
1	0	1
1	1	1

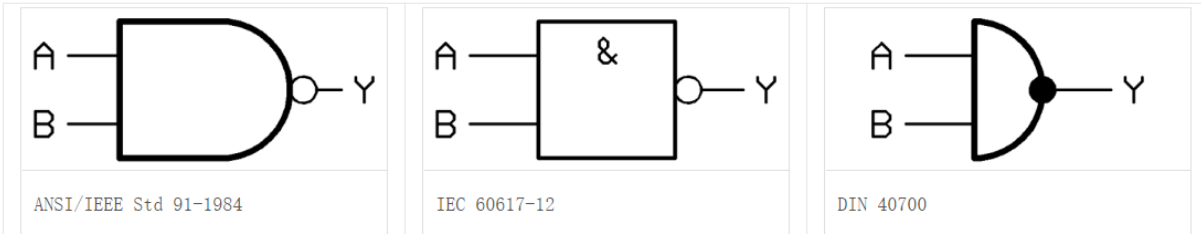


ANSI/IEEE Std 91-1984　IEC 60617-12　DIN符号（DIN 40700）



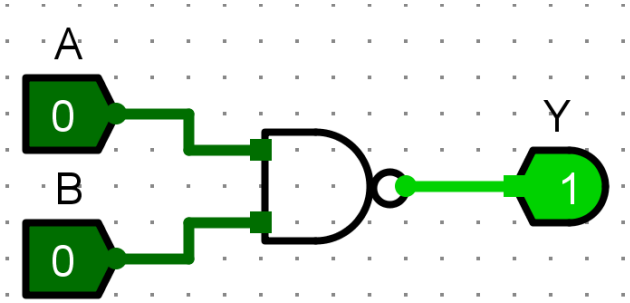
与非门（NAND Gate）

- 与非门：$Y=\overline{A \cdot B} = \overline{A}+\overline{B}$

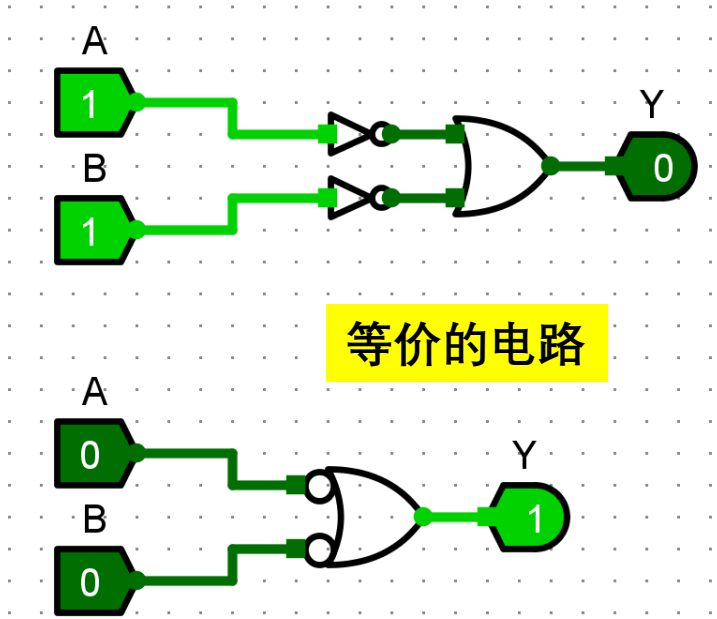


真值表

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

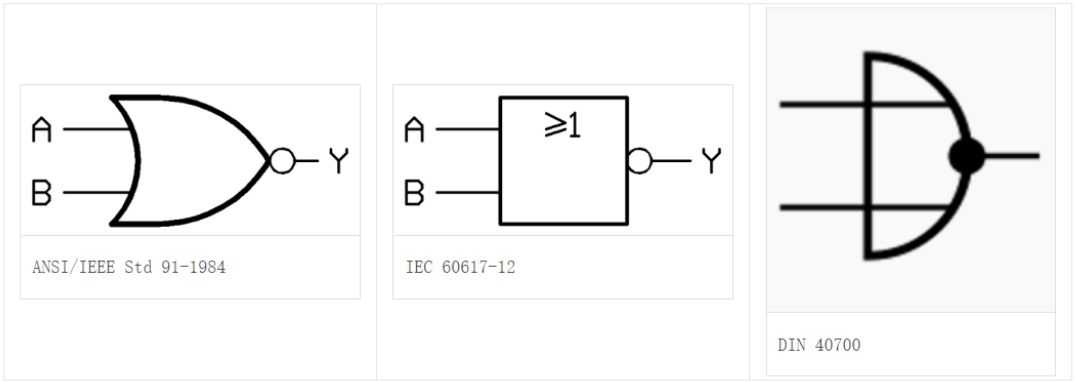


与非门



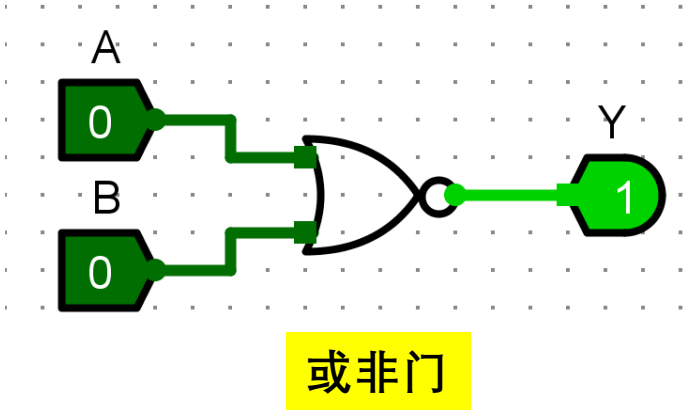
或非门（NOR Gate）

- 或非门：$Y=\overline{A+B} = \overline{A}\cdot\overline{B}$

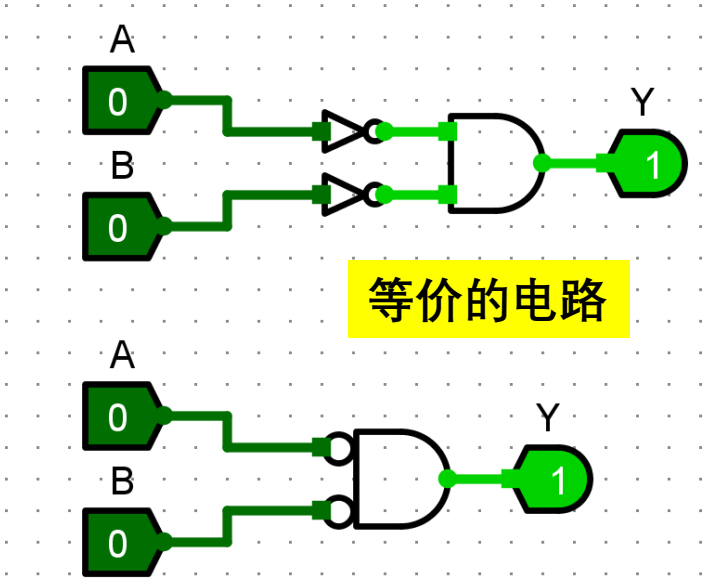


真值表

输入A	输入B	输出Y
0	0	1
0	1	0
1	0	0
1	1	0



或非门



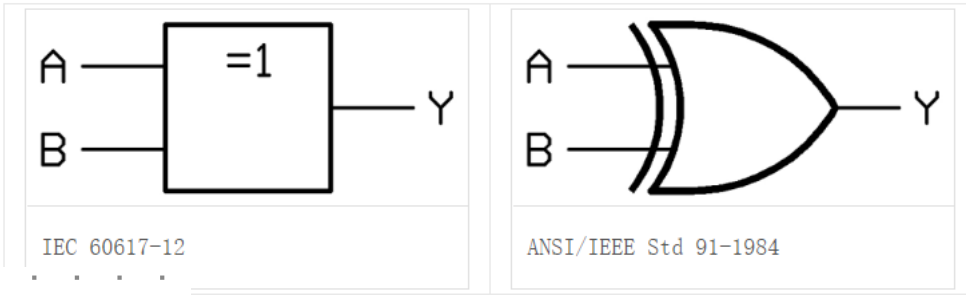
异或门（XOR Gate）

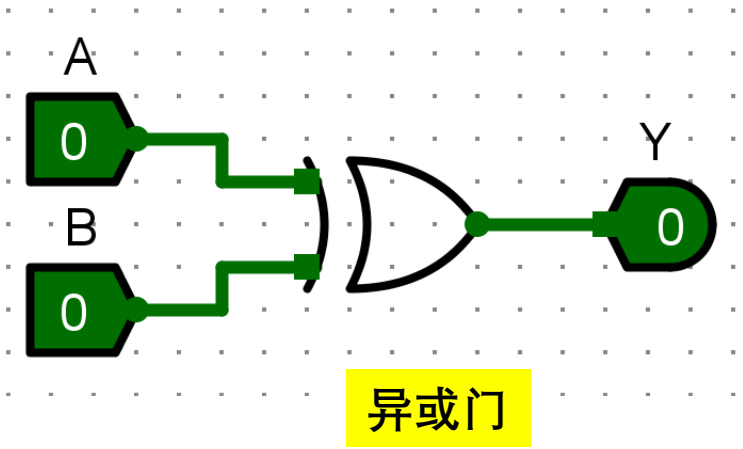
- 异或门：$Y=A \oplus B = \overline{A} \cdot B + A \cdot \overline{B}$

真值表

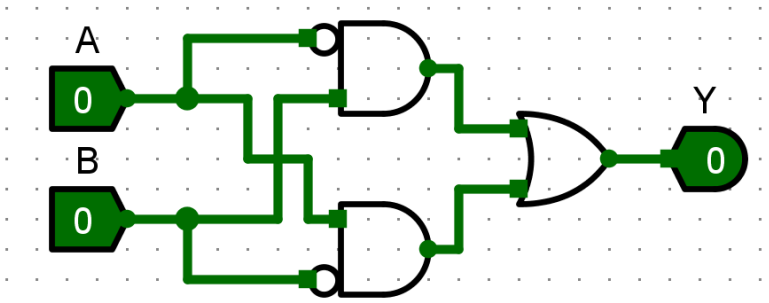
A	B	输出Y
0	0	0
0	1	1
1	0	1
1	1	0

图1





异或门



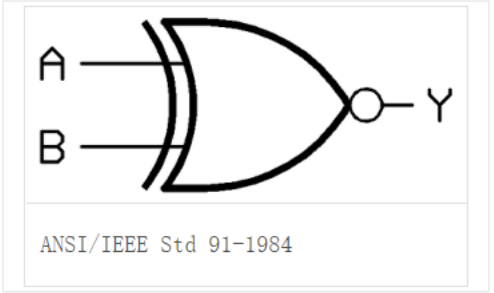
等价的电路

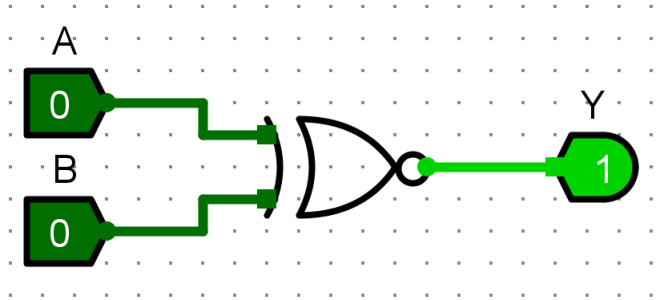
异或非门（XNOR Gate）

- 异或非门（同或门）：$Y=\overline{A \oplus B} = A \cdot B + \overline{A} \cdot \overline{B}$

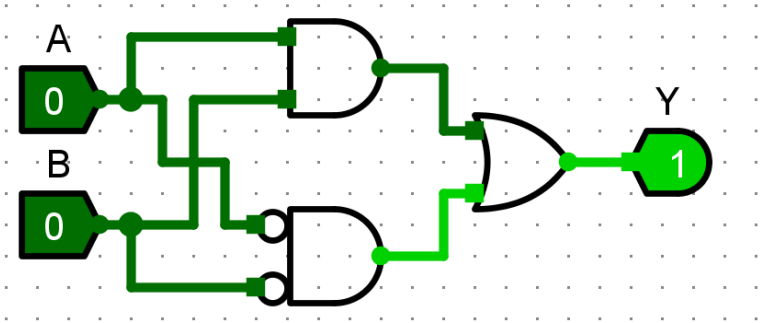
真值表

输入A	输入B	输出F
0	0	1
0	1	0
1	0	0
1	1	1



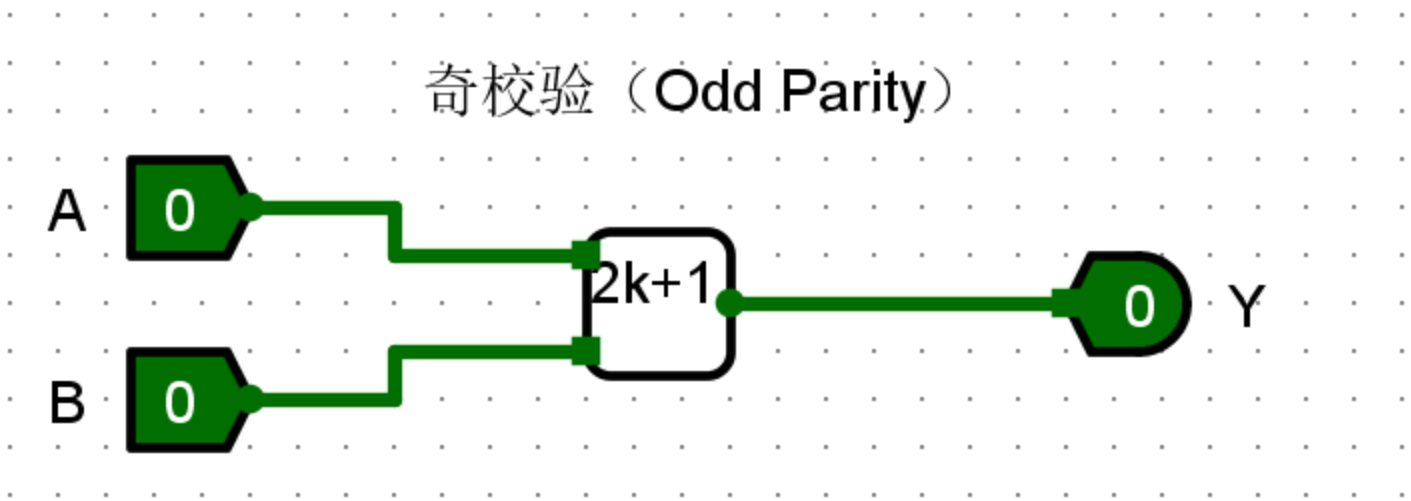


异或非门（同或门）



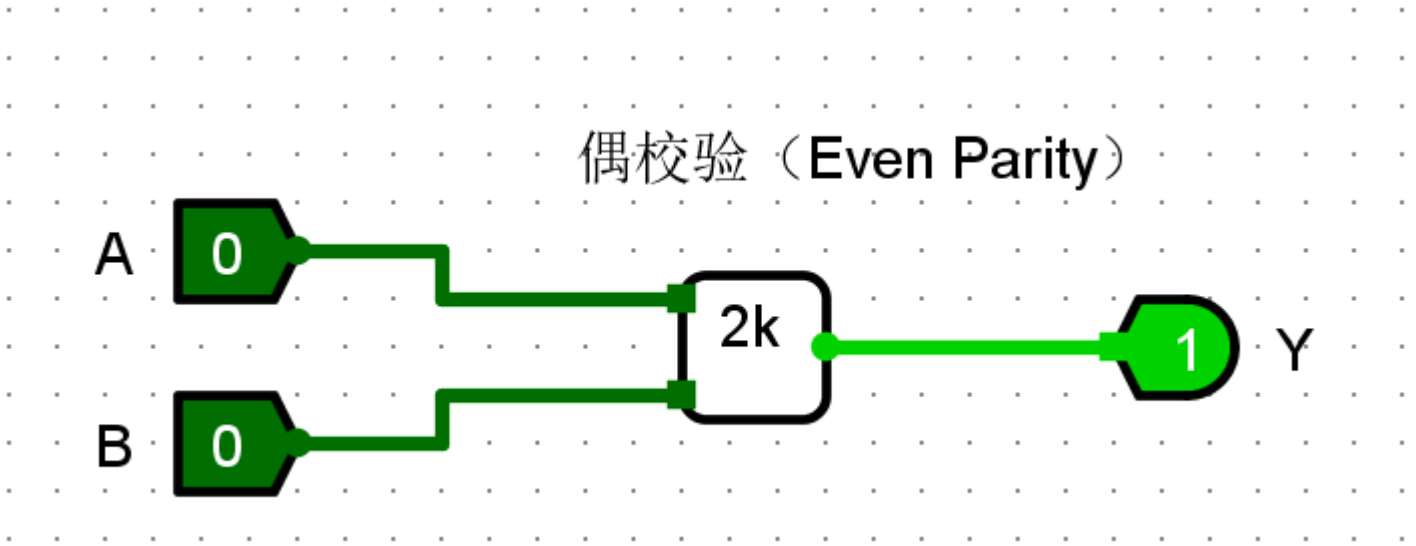
等价的电路

奇校验（Odd Parity）



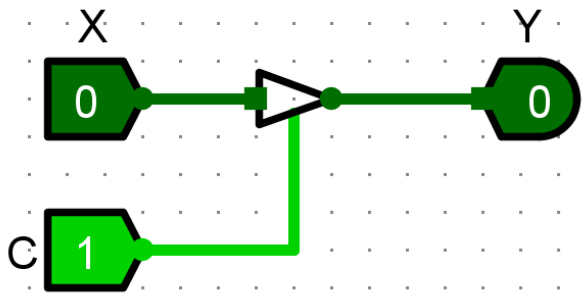
Y = A 异或 B

偶校验（Even Parity）

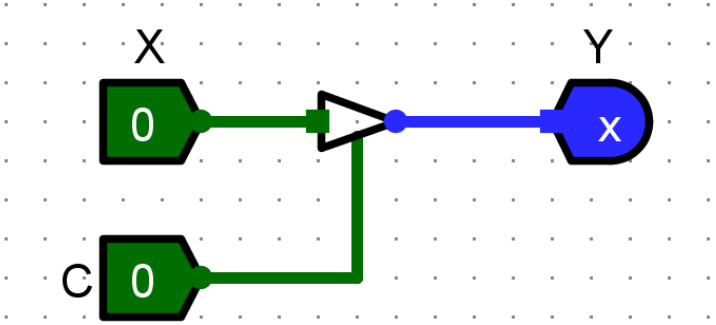


Y = A 异或非 B

可控缓冲器（Controlled Buffer）

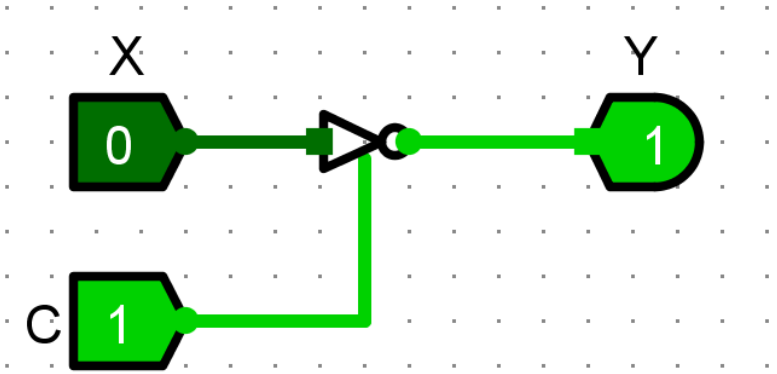


C=1　Y=X

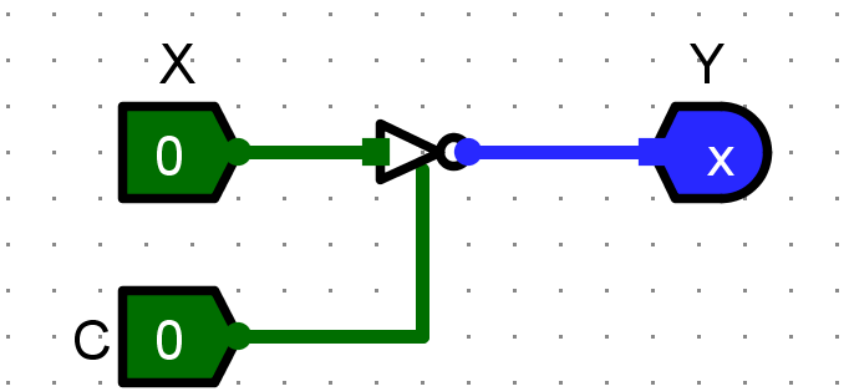


C=0　Y=高阻

可控反相器（Controlled Inverter）

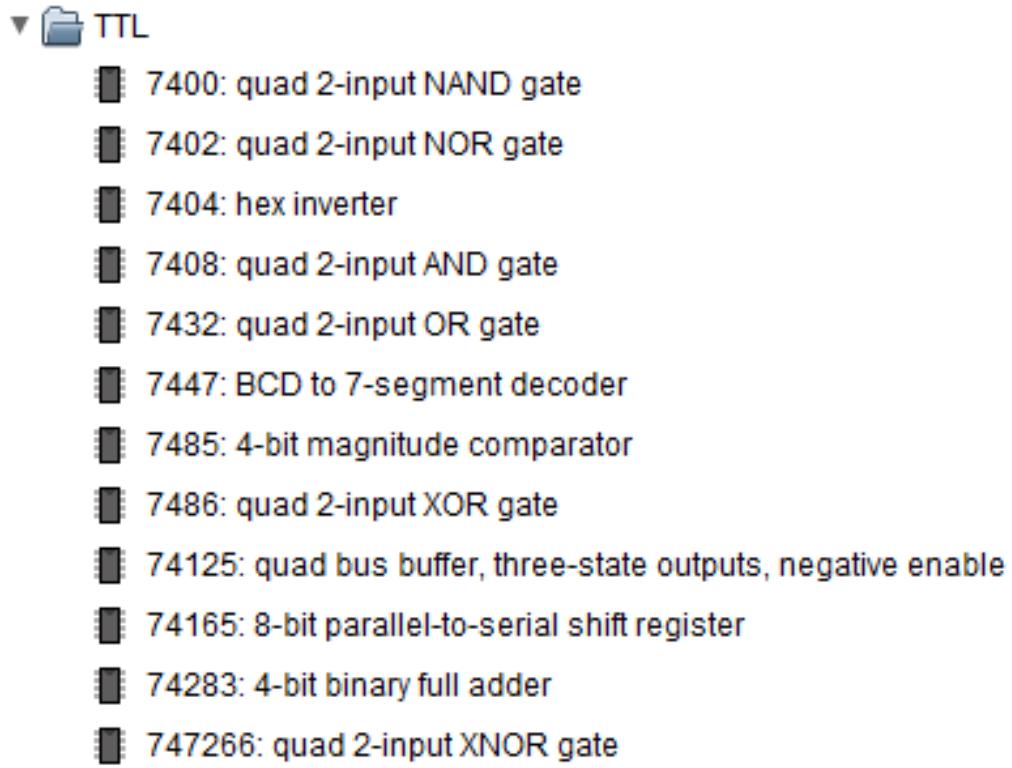


C=1　Y=/X

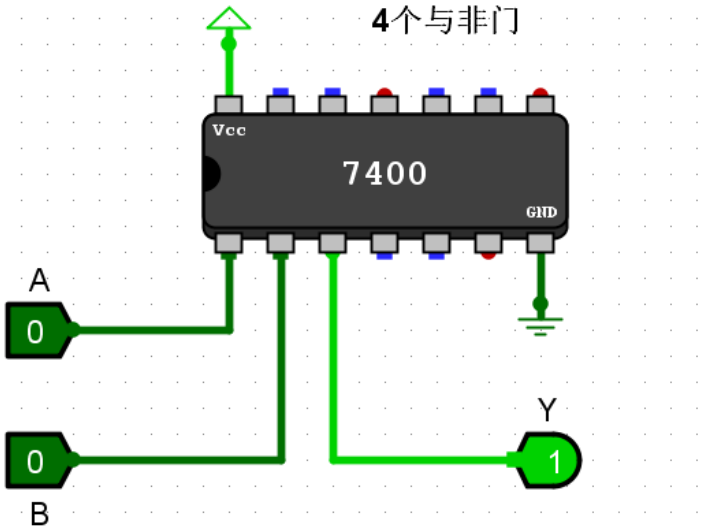


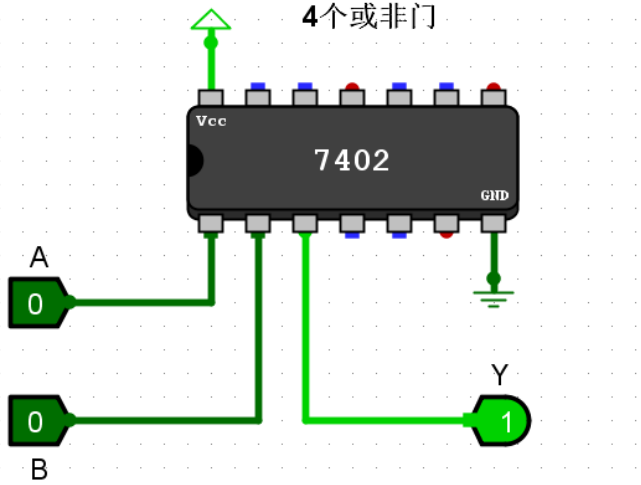
C=0　Y=高阻

（三）TTL芯片

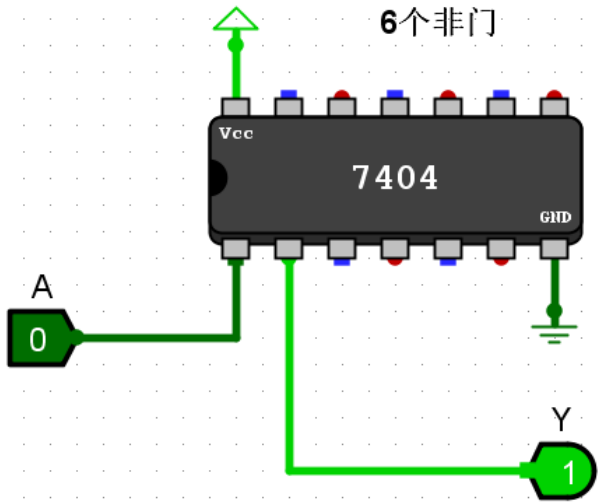

TTL
7400: quad 2-input NAND gate
7402: quad 2-input NOR gate
7404: hex inverter
7408: quad 2-input AND gate
7432: quad 2-input OR gate
7447: BCD to 7-segment decoder
7485: 4-bit magnitude comparator
7486: quad 2-input XOR gate
74125: quad bus buffer, three-state outputs, negative enable
74165: 8-bit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

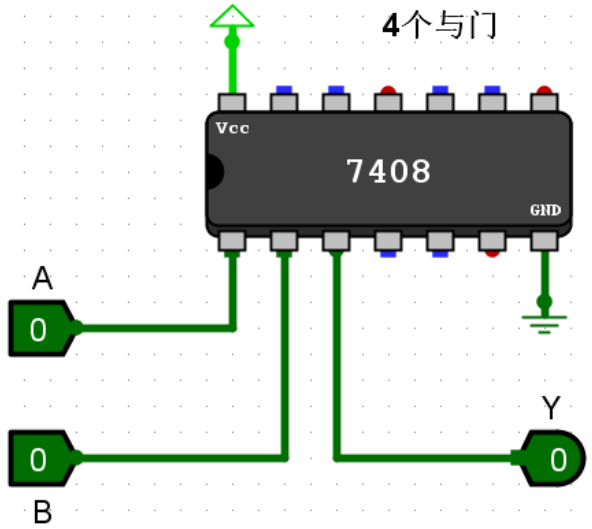
7400、7402



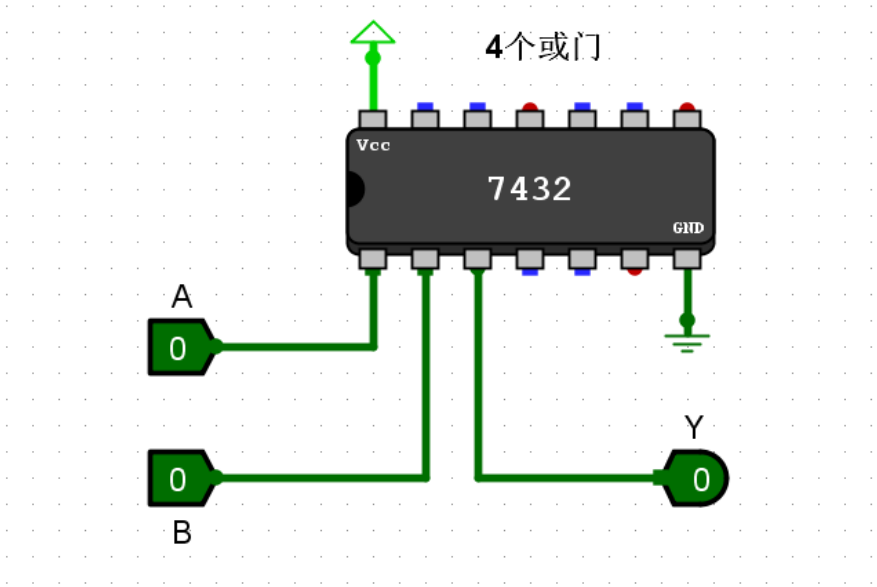


7404、7408

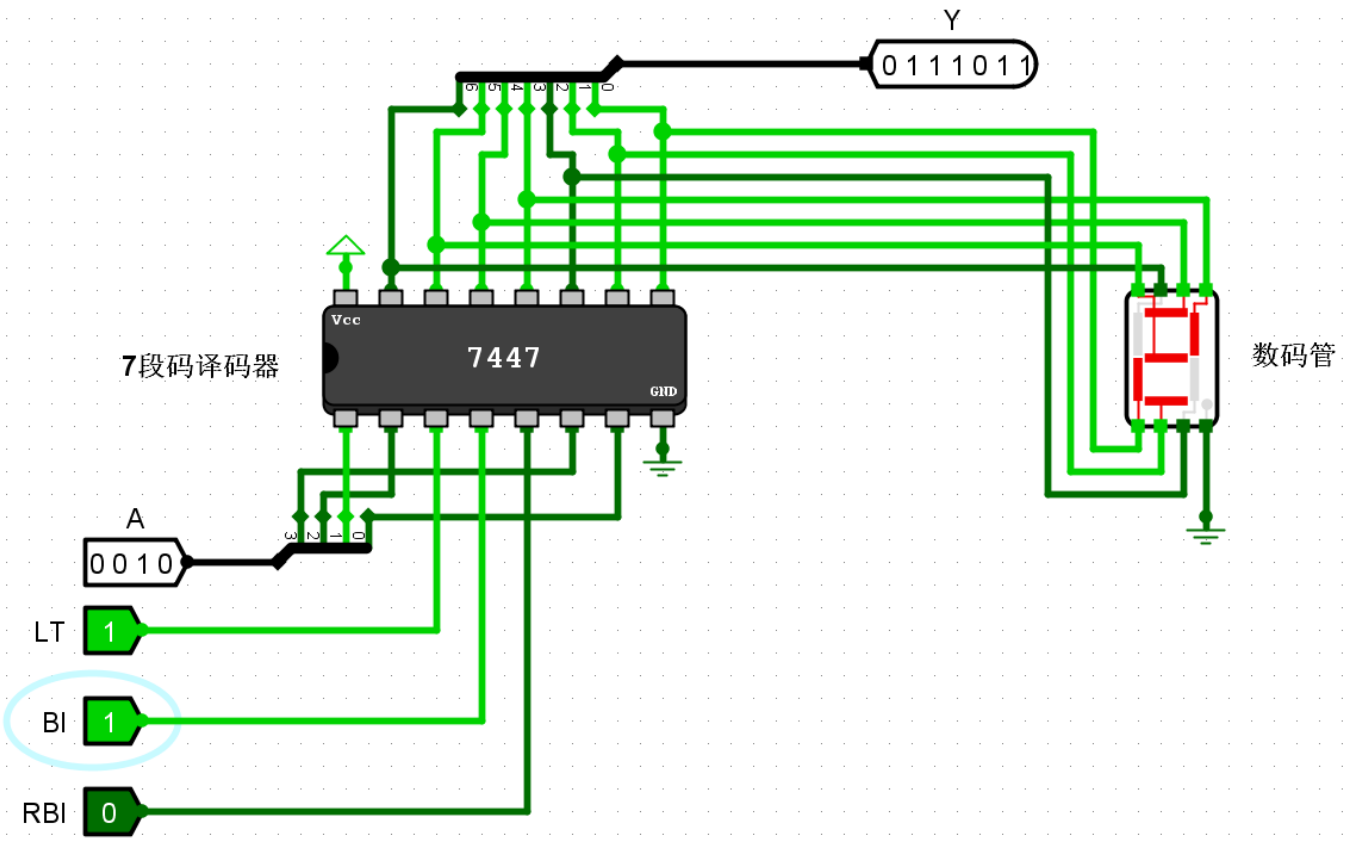




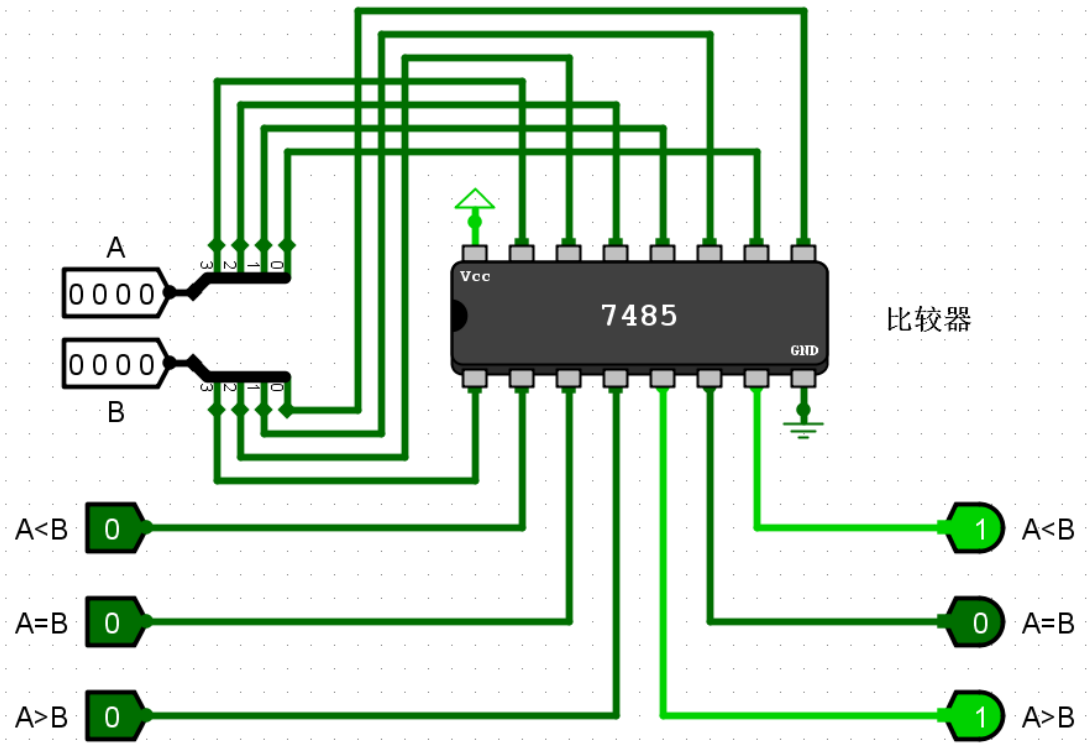
7432



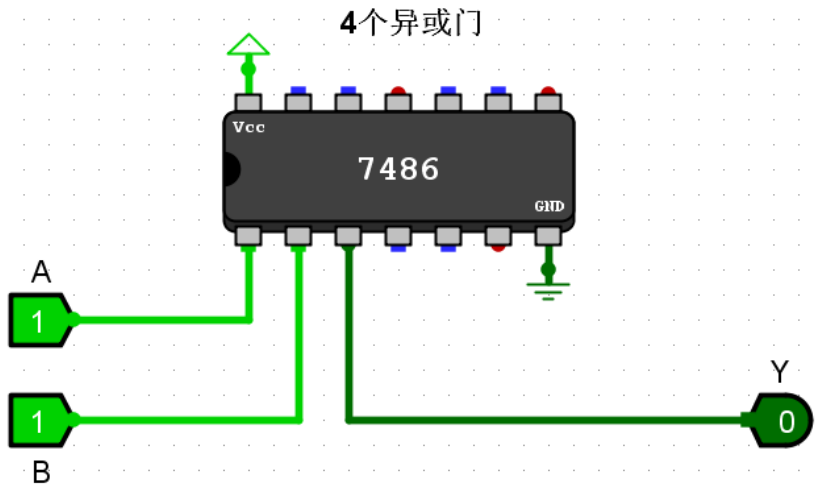
7447

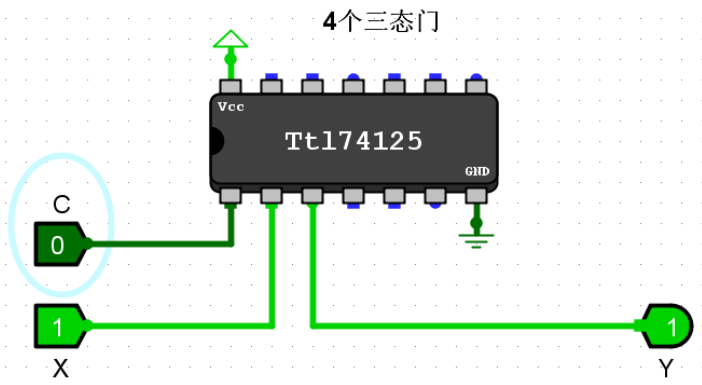


7485



7486、74125

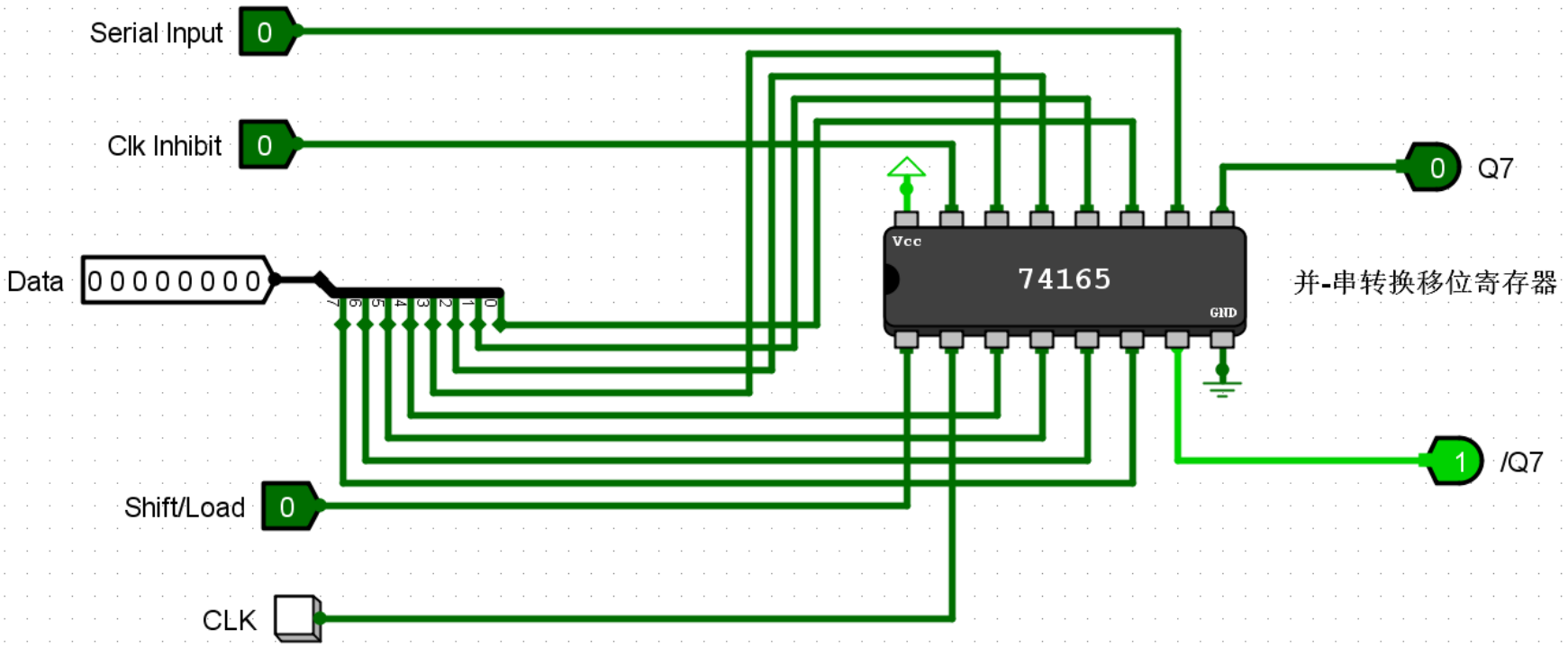




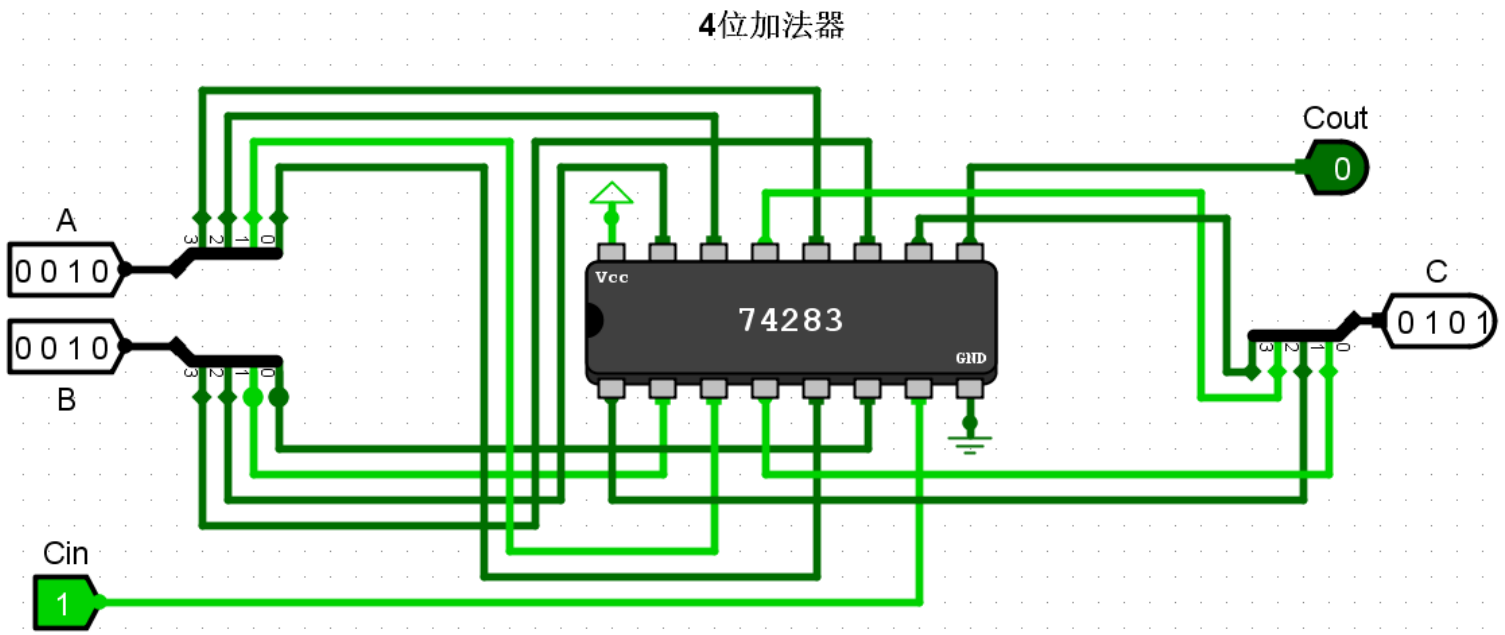
C=0　Y=X

C=1　Y=高阻

74165

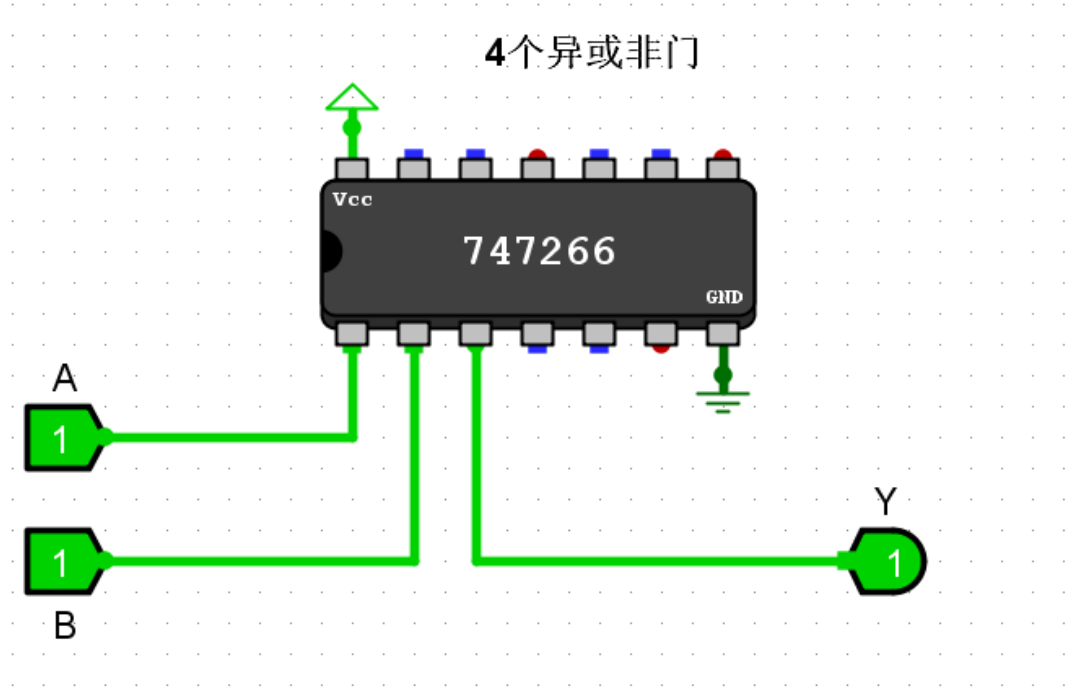

Serial Input
0
Clk Inhibit
0
Data
0 0 0 0 0 0 0 0
Shift/Load
0
CLK
Vcc
74165
GND
并-串转换移位寄存器
0
Q7
1
/Q7

74283

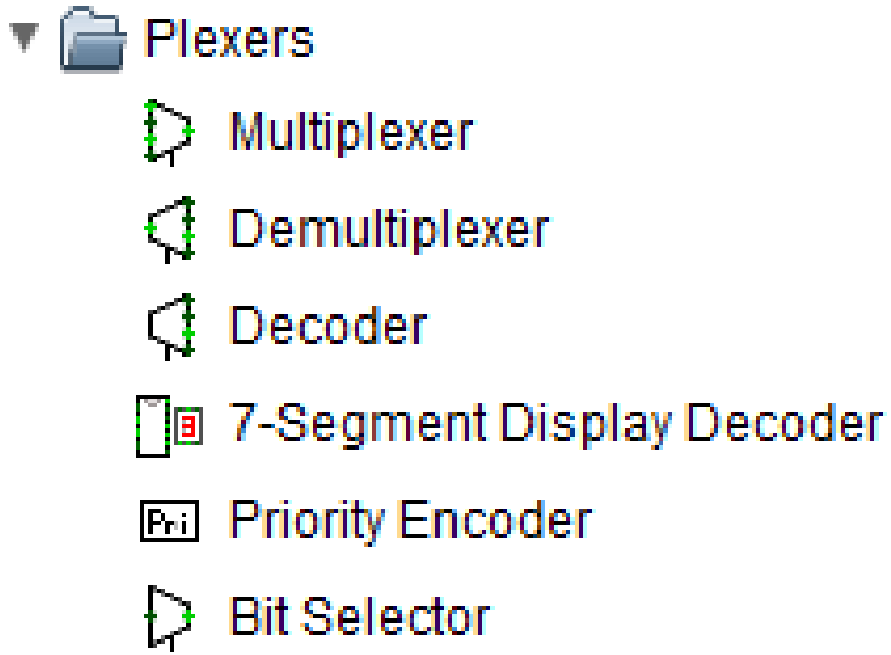


{Cout, C} = A+B+Cin

74266

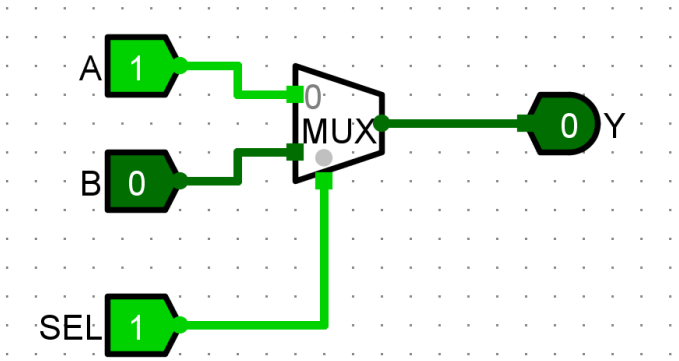


（四）多路复用器（Plexers）



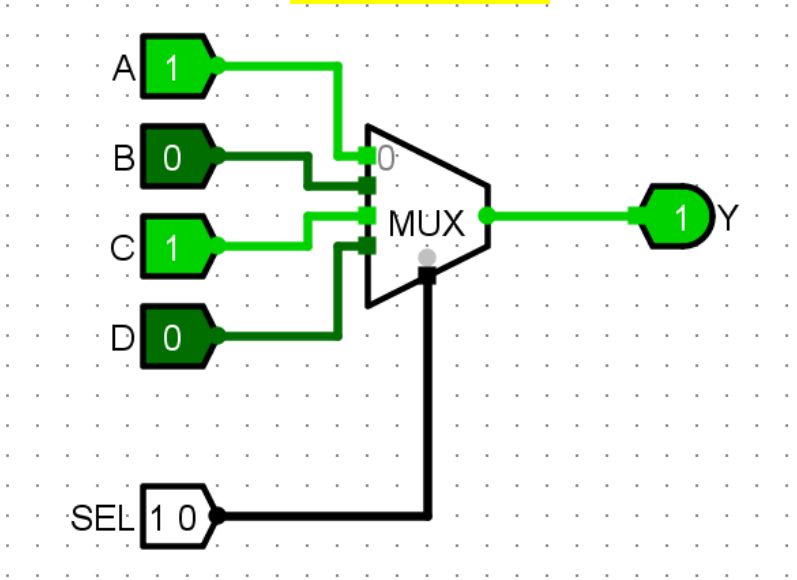
多路选择器（Multiplexer）

2路选择器



SEL=0　Y=A
SEL=1　Y=B

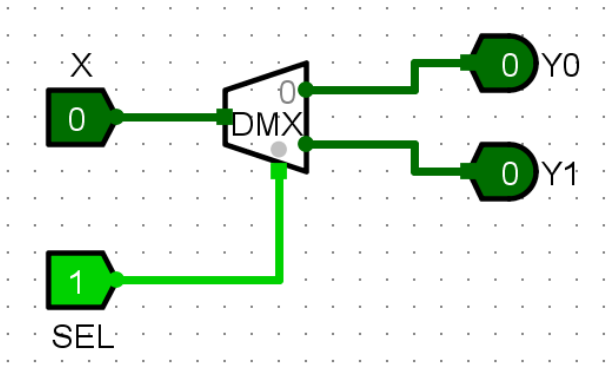
4路选择器



SEL=00　Y=A
SEL=01　Y=B
SEL=10　Y=C
SEL=11　Y=D

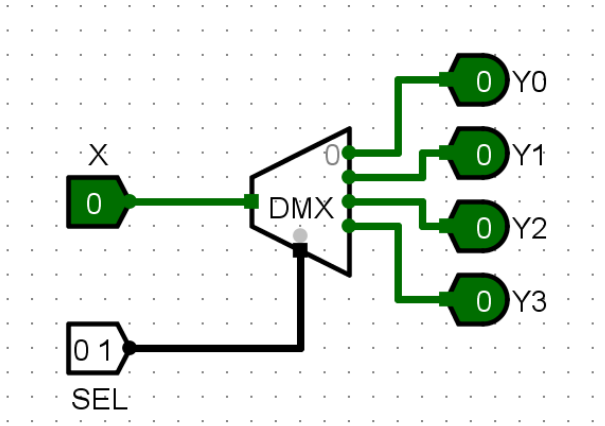
多路分配器（Demultiplexer）

2路分配器



SEL=0　Y0=X
SEL=1　Y1=X

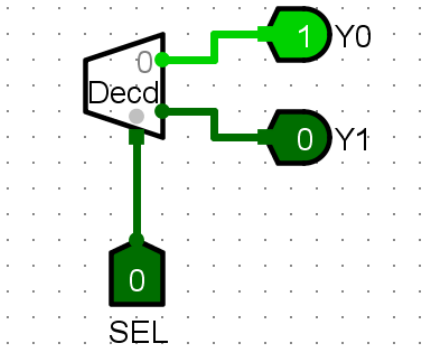
4路分配器



SEL=00　Y0=X
SEL=01　Y1=X
SEL=10　Y2=X
SEL=11　Y3=X

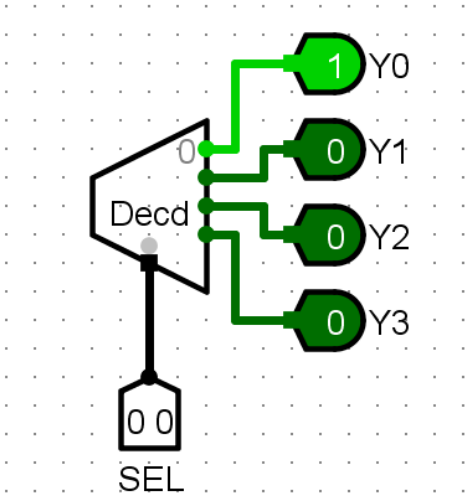
解码器（Decoder）

1位解码器



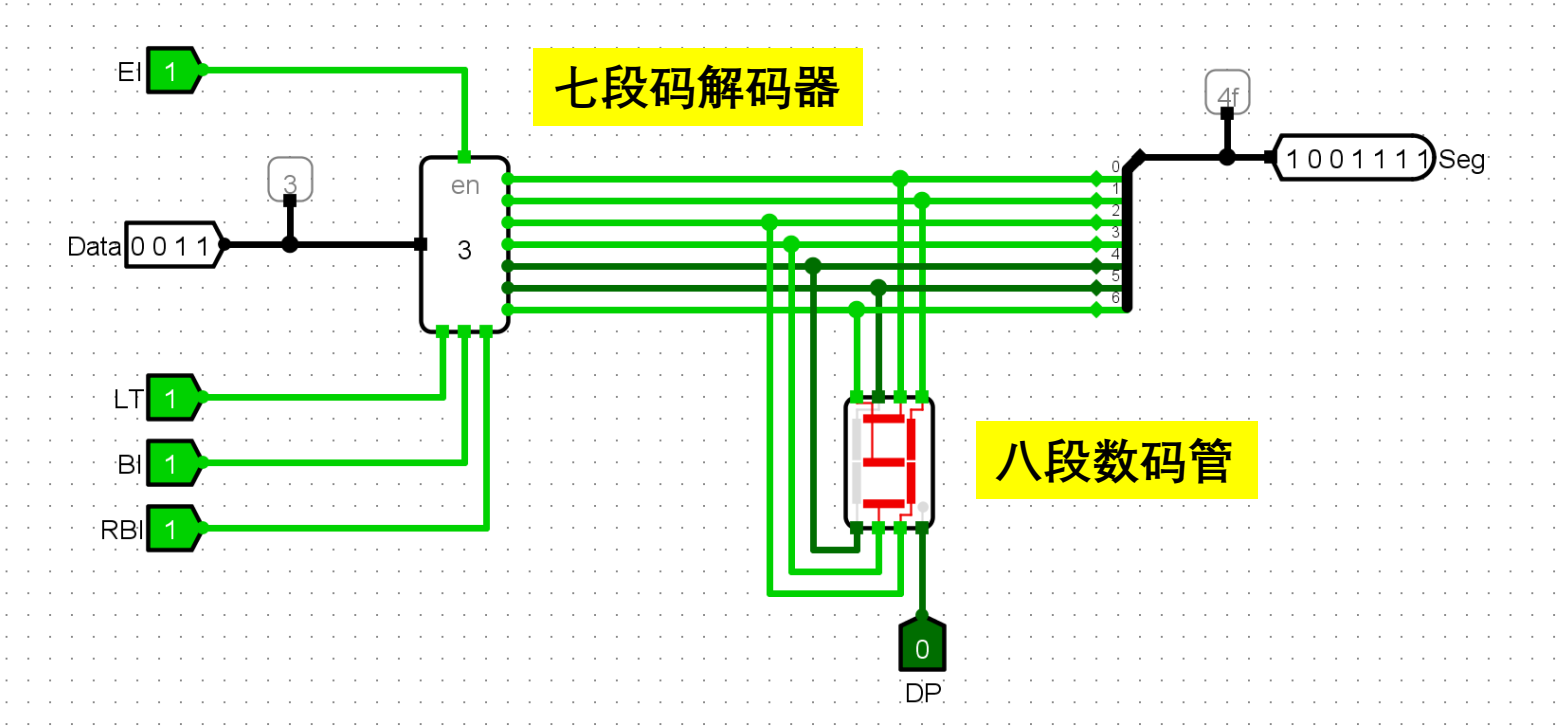
SEL=0	Y0=1	Y1=0
SEL=1	Y0=0	Y1=1

2位解码器

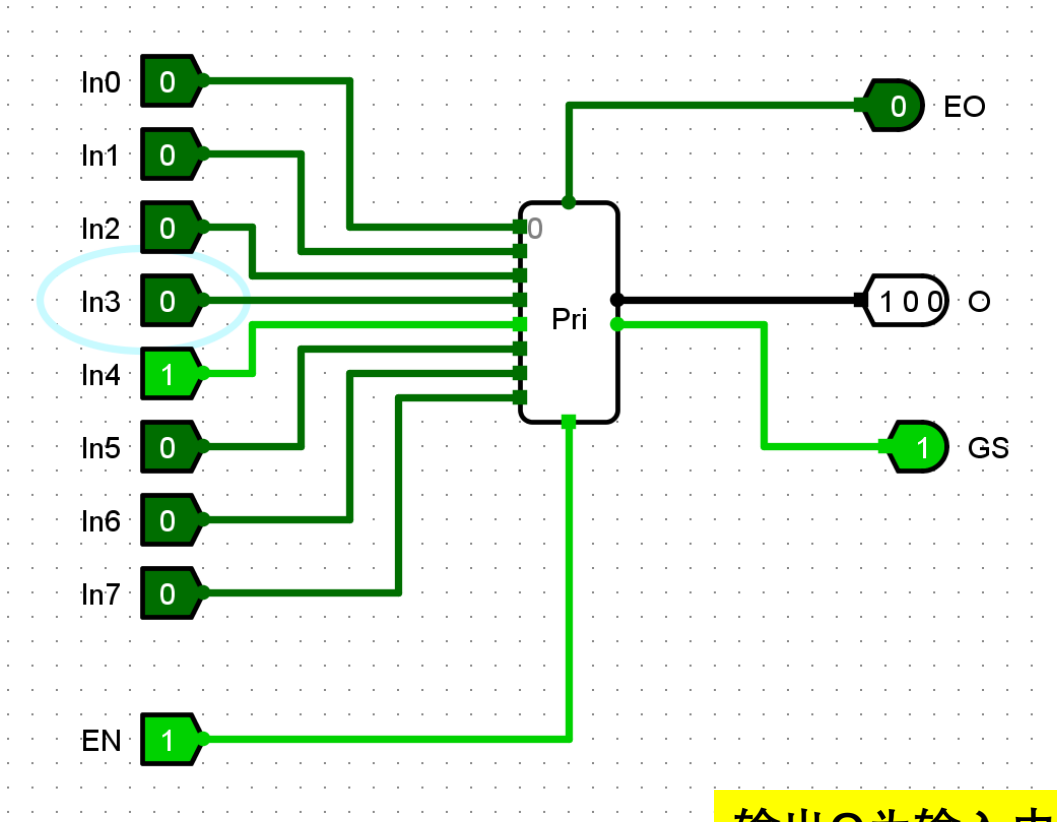


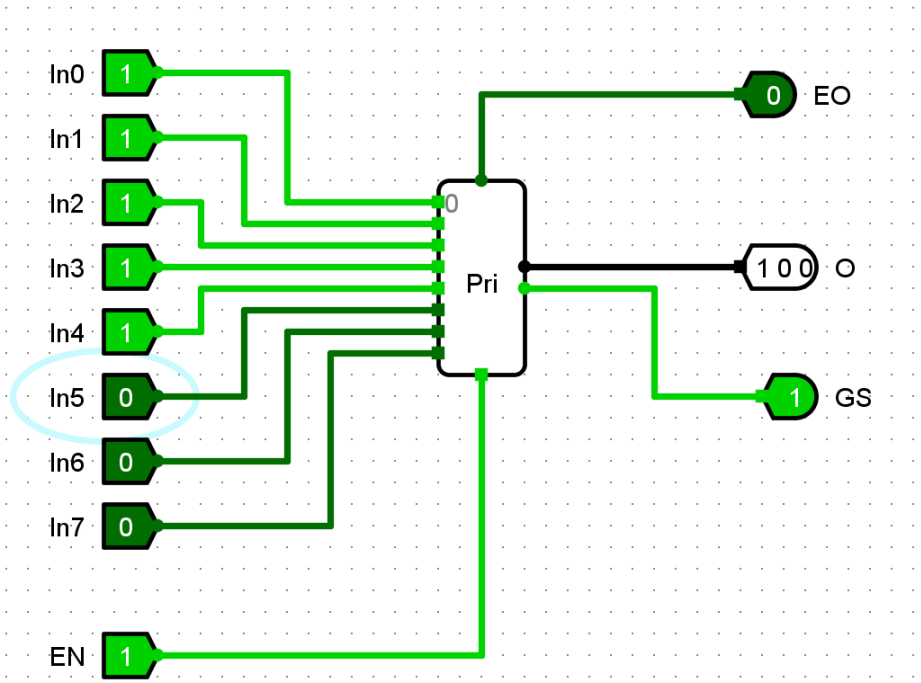
SEL=00	Y0=1	Y1=0	Y2=0	Y3=0
SEL=01	Y0=0	Y1=1	Y2=0	Y3=0
SEL=10	Y0=0	Y1=0	Y2=1	Y3=0
SEL=11	Y0=0	Y1=0	Y2=0	Y3=1

七段码解码器（7-Segment Display Decoder）



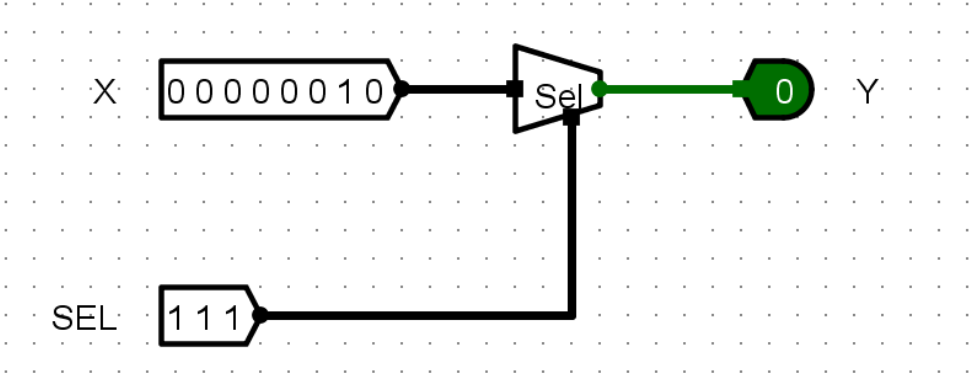
优先级编码器（Priority Encoder）





输出O为输入中最高位1的位置

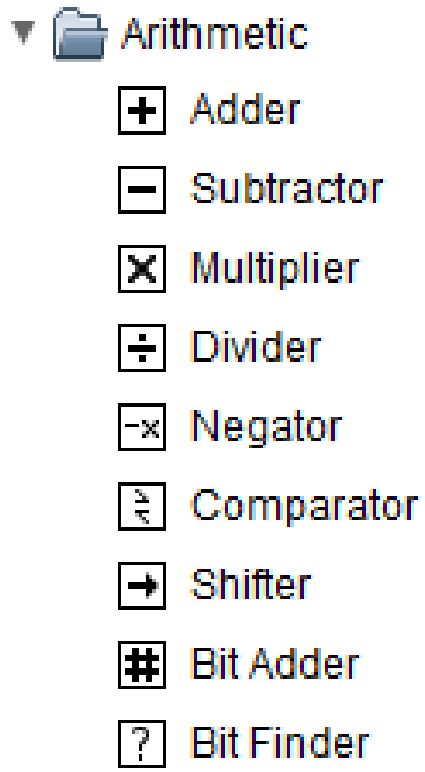
位选择器（Bit Selector）



$X=X_7X_6X_5X_4X_3X_2X_1X_0$

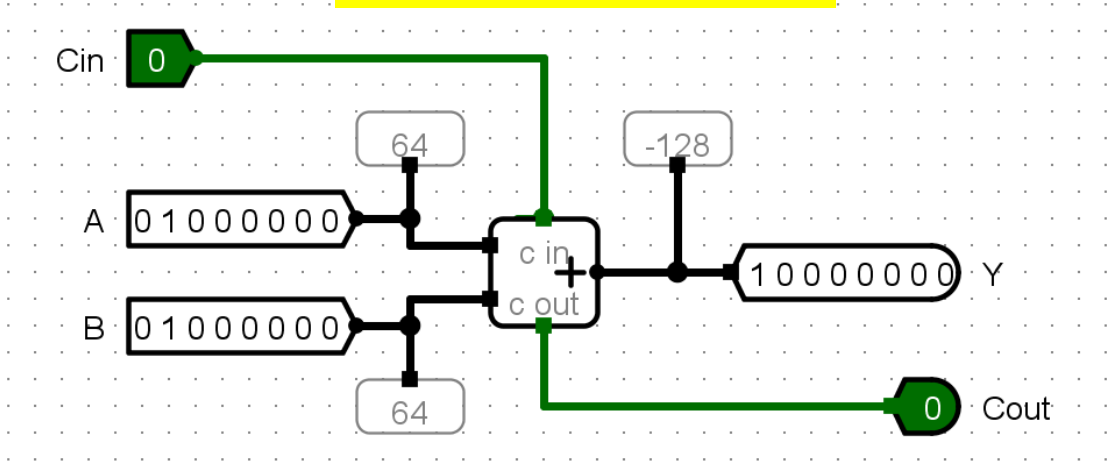
SEL=000　$Y=X_0$
SEL=001　$Y=X_1$
SEL=010　$Y=X_2$
SEL=011　$Y=X_3$
SEL=000　$Y=X_4$
SEL=001　$Y=X_5$
SEL=010　$Y=X_6$
SEL=011　$Y=X_7$

（五）算术运算部件（Arithmetic）



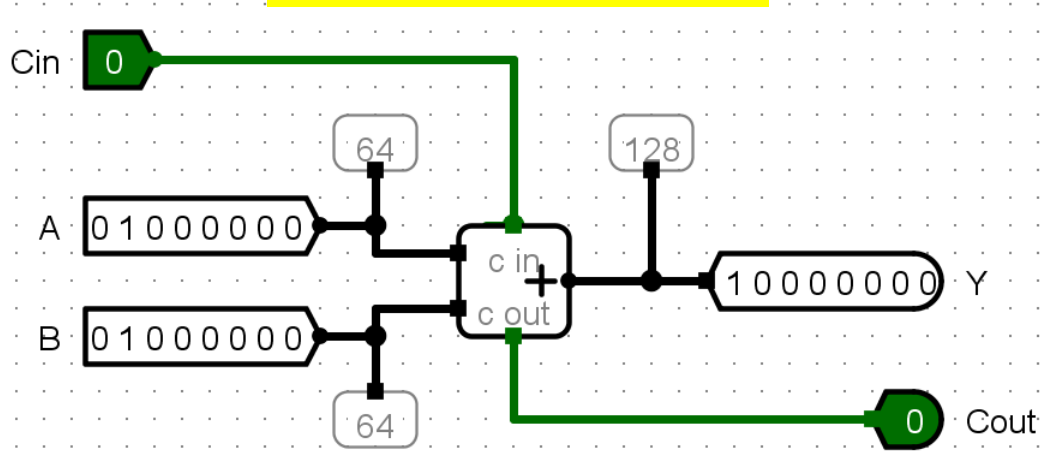
加法器（Adder）

有符号数加法（8位）



{ Cout, Y } = A + B + Cin

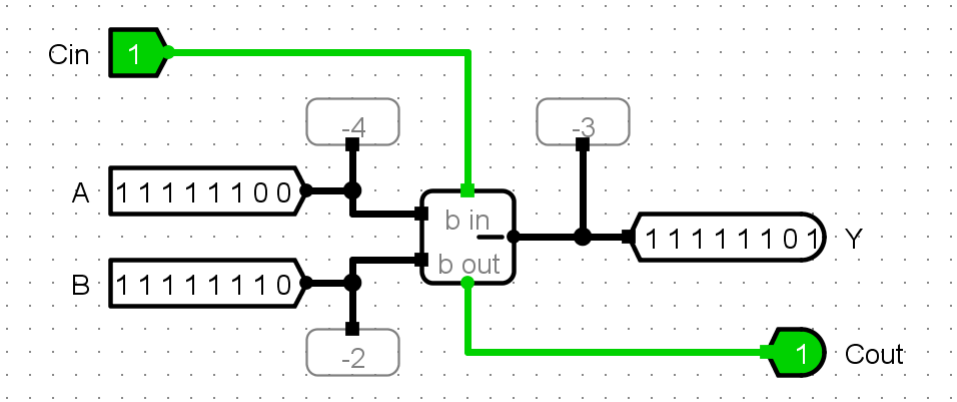
无符号数加法（8位）



{ Cout, Y } = A + B + Cin

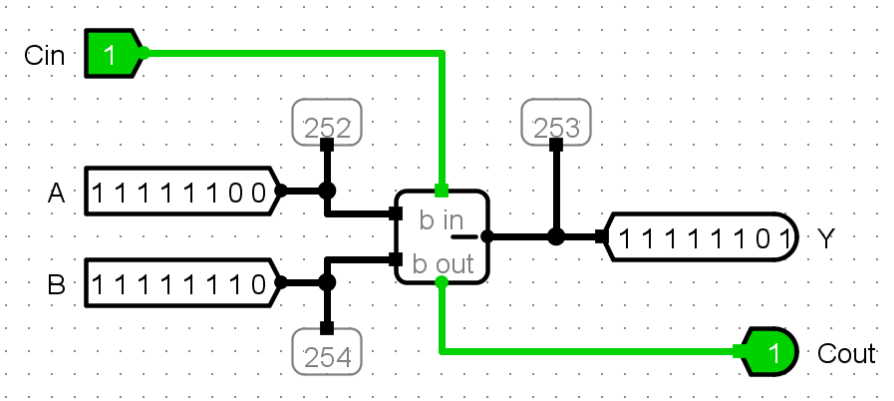
减法器（Subtractor）

有符号数减法（8位）



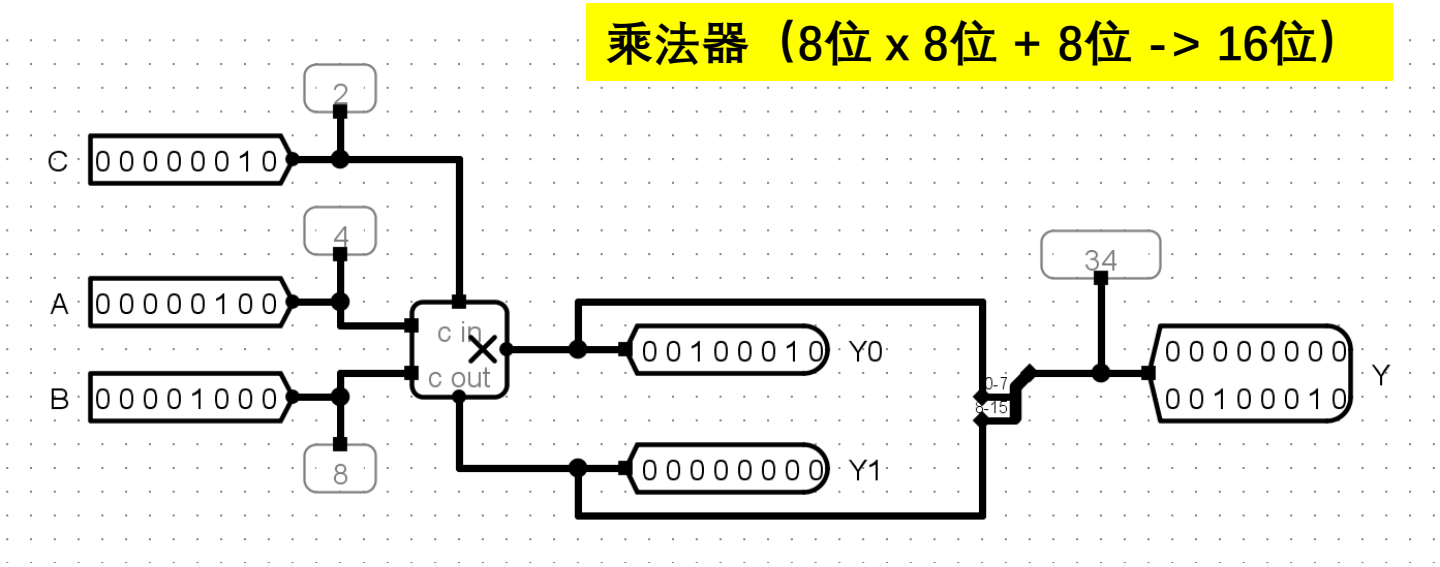
{ Cout, Y } = A - B - Cin

无符号数减法（8位）



{ Cout, Y } = A - B - Cin

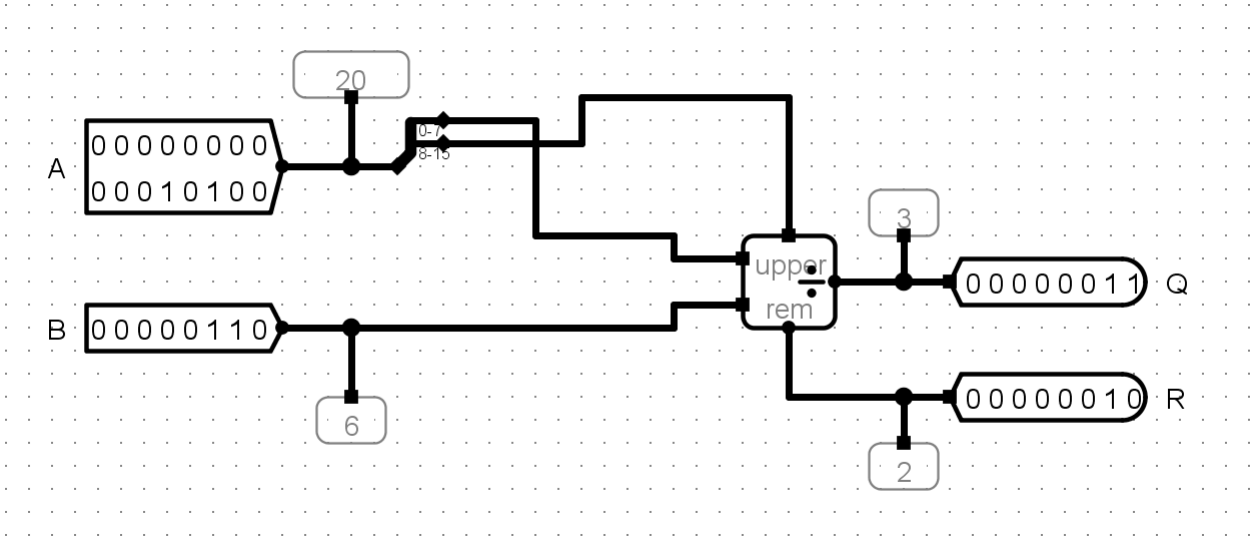
乘法器（Multiplier）



Y = { Y1, Y0 } = { A x B + C }

除法器（Divider）

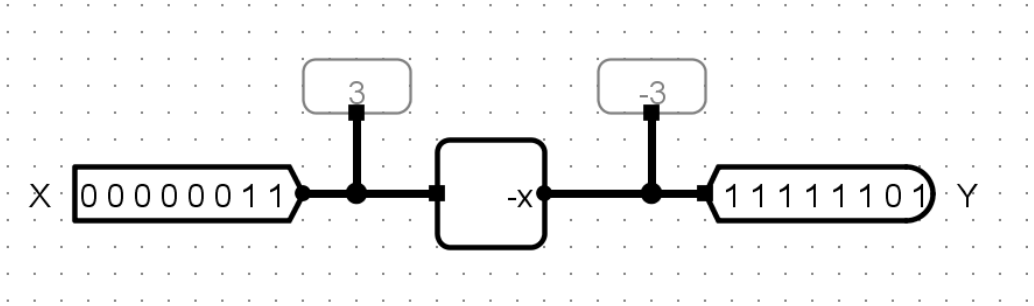
除法器（16位 ÷ 8位　商Q=8位　余数R=8位）

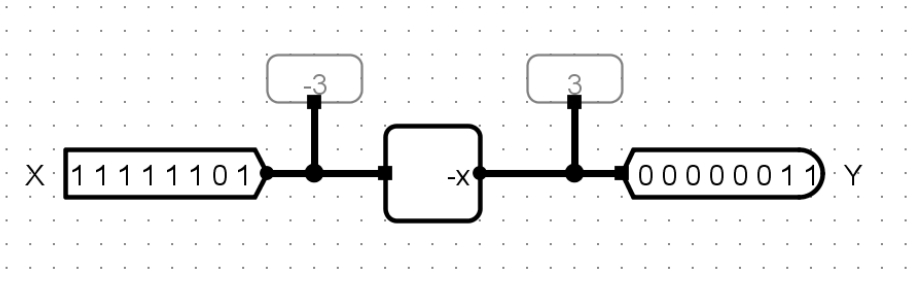


{ Q, R } =　A ÷ B

求补器（Negator）

Y = X的补数（不是补码）　　Y = X的模 - X

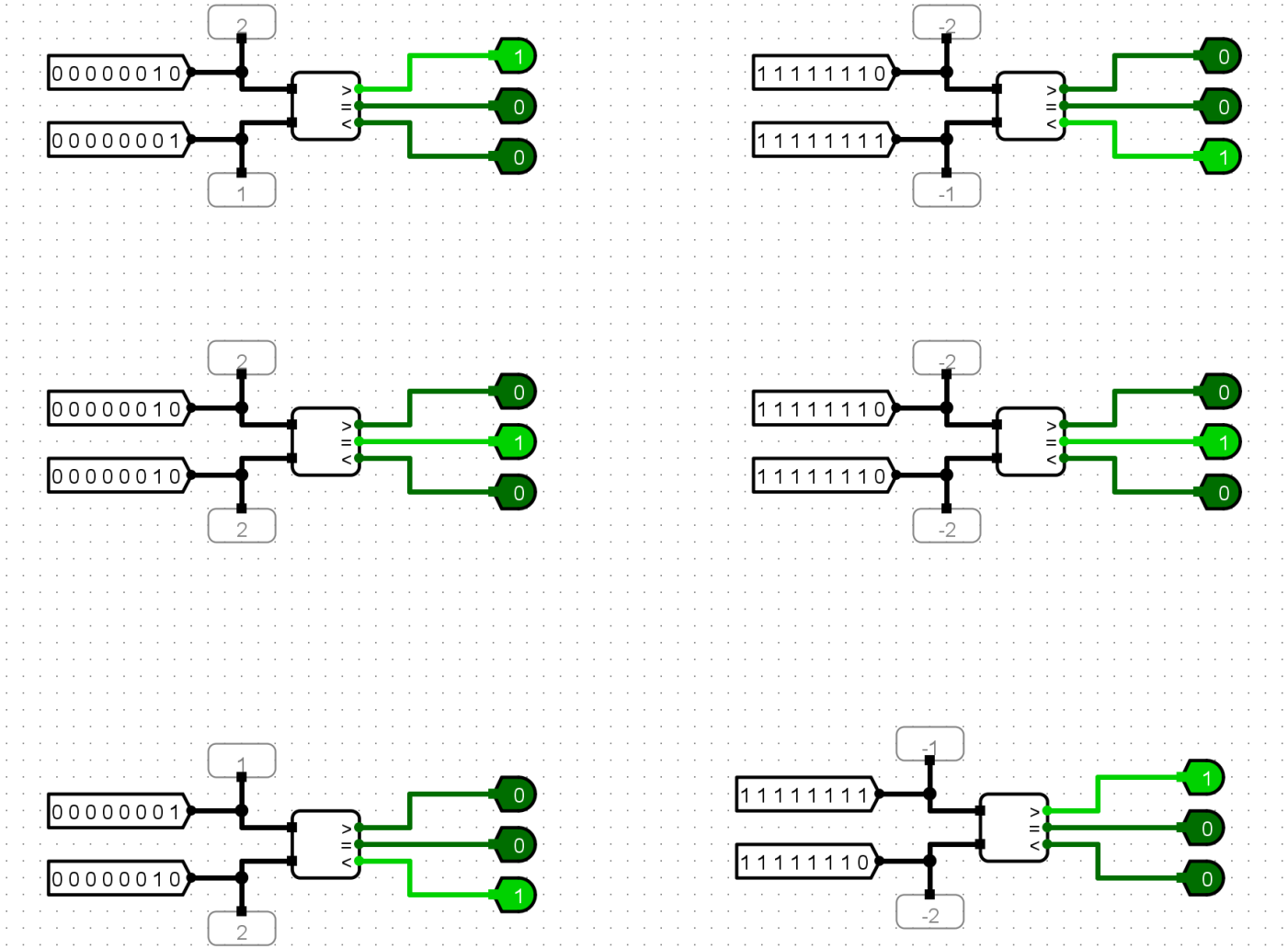




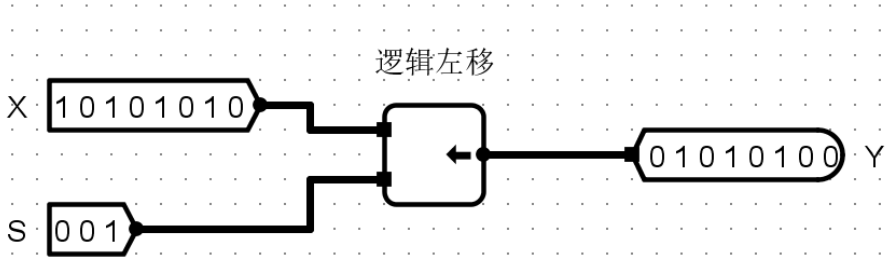
Y = X的模 – X = 2^8 - X

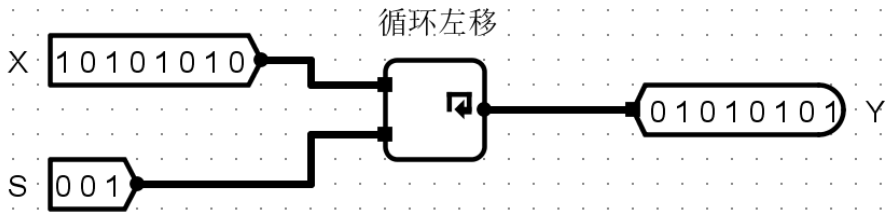
Y = X的模 – X = 2^8 - X

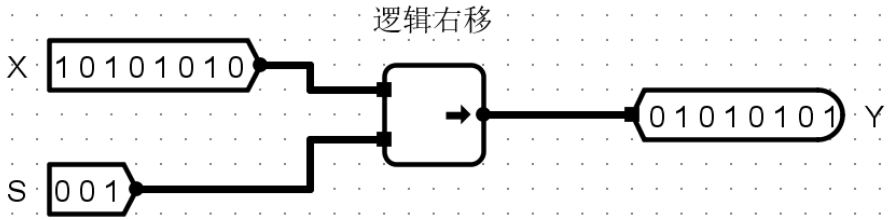
比较器（Comparator）

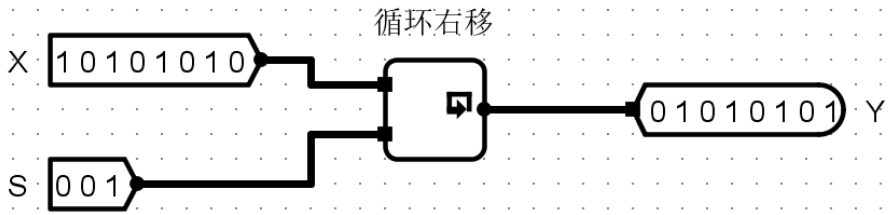


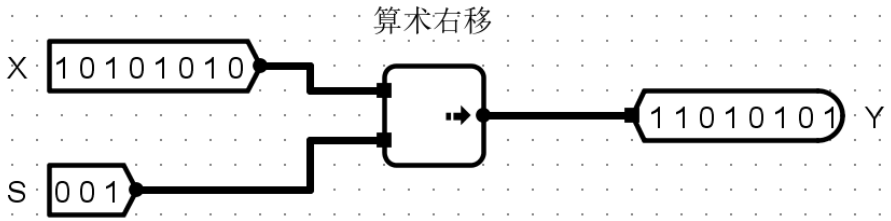
移位器（Shifter）







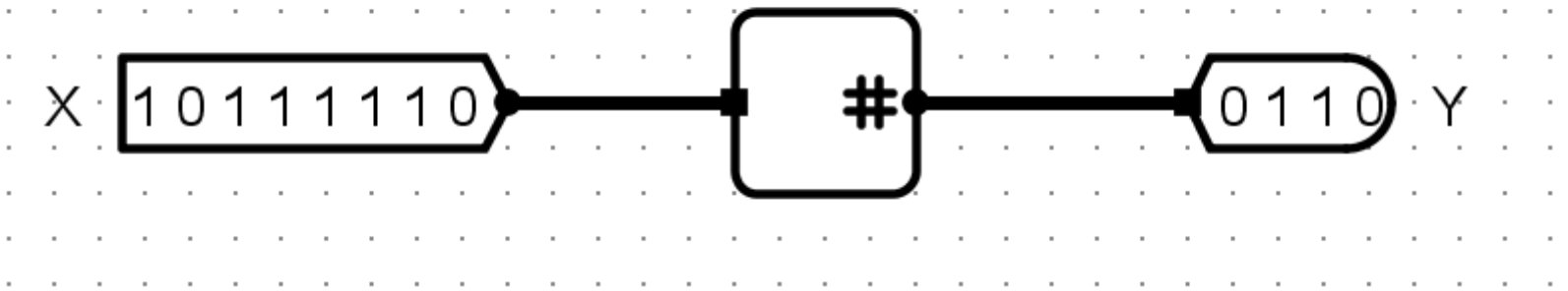




Y = X 移位 S次

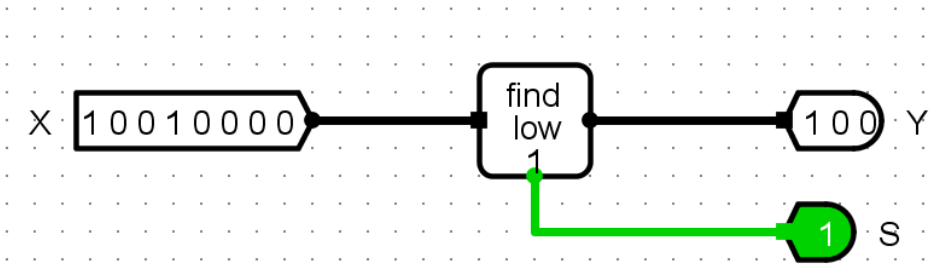
算术左移=逻辑左移

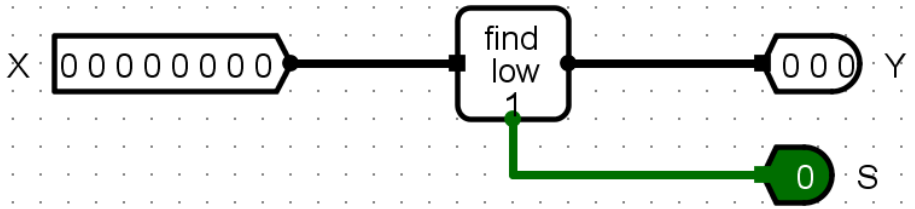
位加法器（Bit Adder）



Y = X中1的个数

位发现器（Bit Finder）





如果X≠0，则Y=X中从右开始第1个“1”的位置

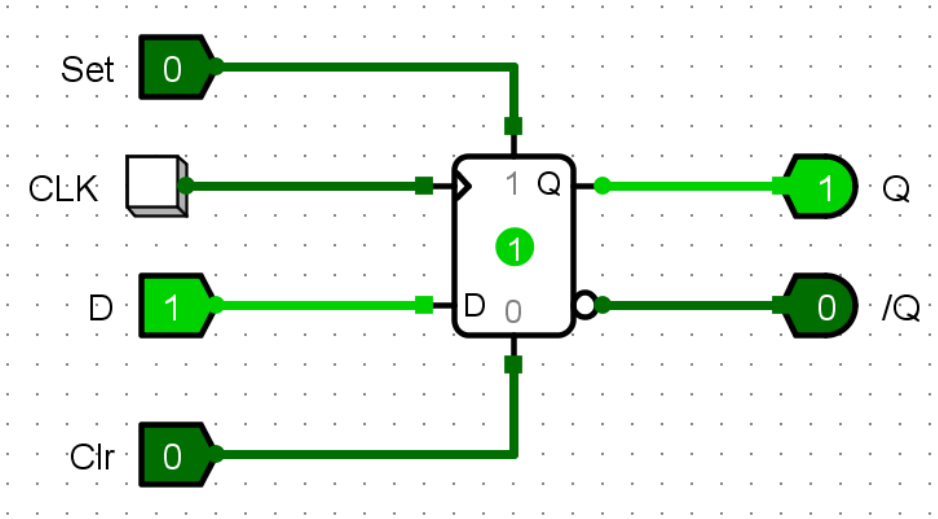
并且置S=1

最右边位置=0，最左边位置=7

如果X=0，则Y=000，并且置S=0

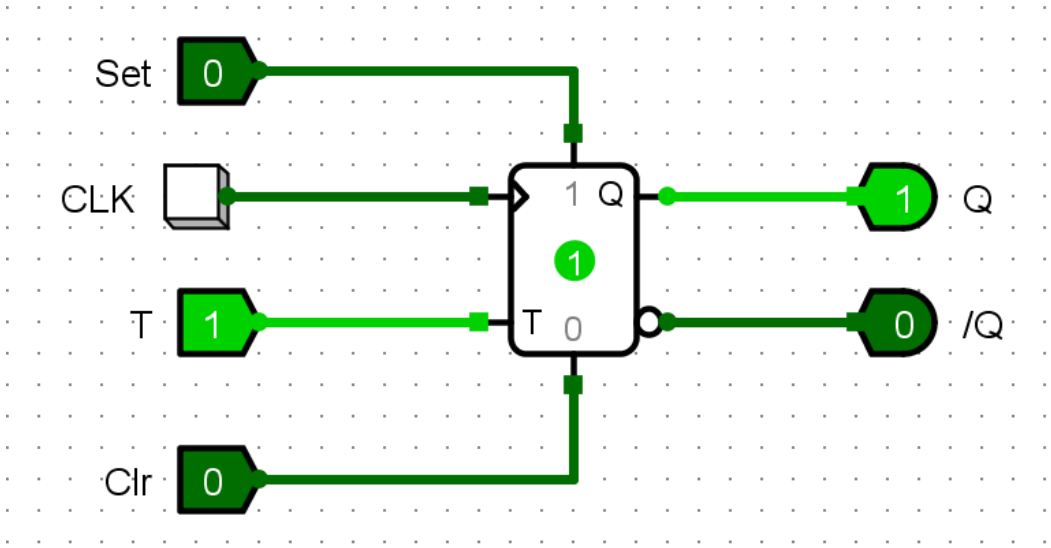
（六）存储单元（Memory）

D触发器（D Flip-Flop）



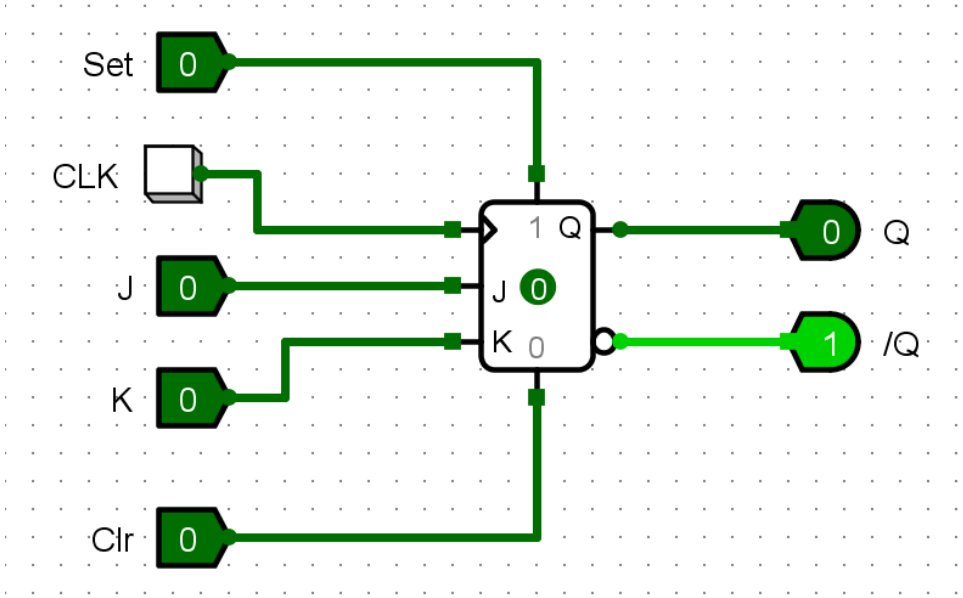
输入				输出	
CLK	Set	Clr	D	Q	/Q
上升沿	0	0	0	0	1
上升沿	0	0	1	1	0
x	0	1	x	0	1
x	1	0	x	1	0
x	1	1	x	0	1

T触发器（T Flip-Flop）



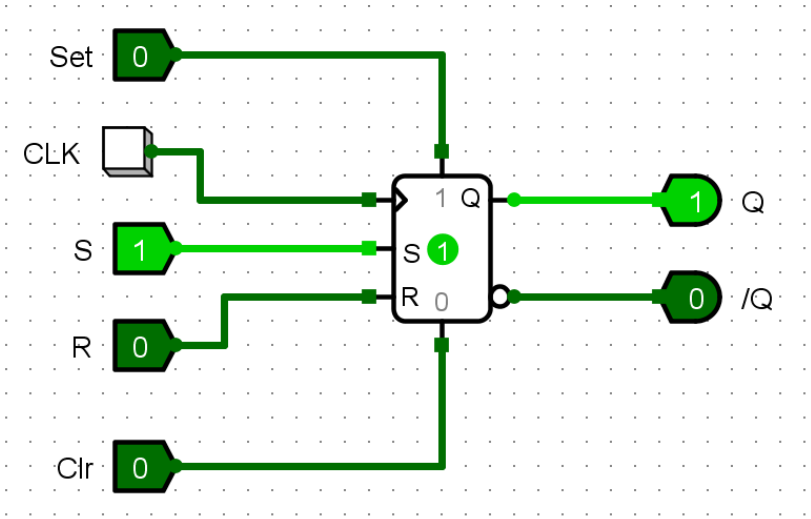
输入				输出	
CLK	Set	Clr	T	Q	/Q
上升沿	0	0	0	保持不变	保持不变
上升沿	0	0	1	翻转	翻转
x	0	1	x	0	1
x	1	0	x	1	0
x	1	1	x	0	1

J-K触发器（J-K Flip-Flop）



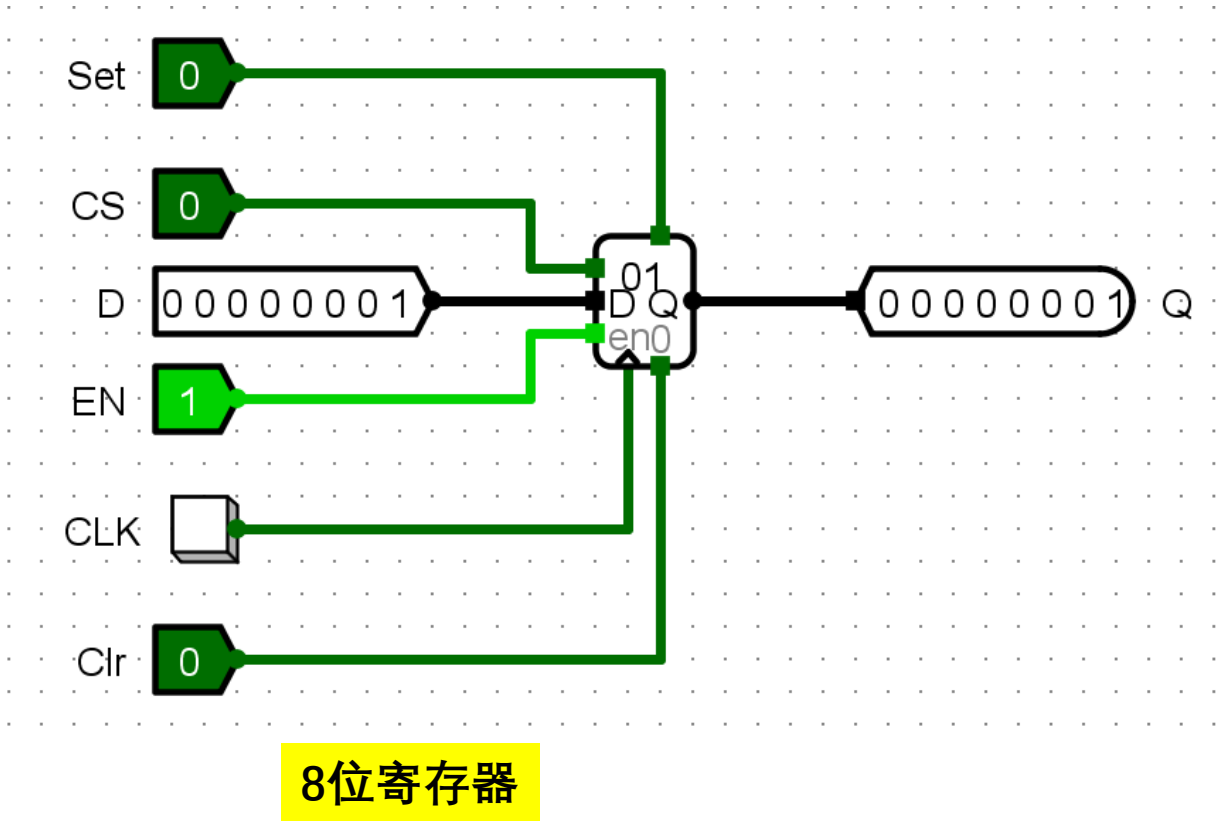
输入					输出	
CLK	Set	Clr	J	K	Q	/Q
上升沿	0	0	0	0	保持不变	保持不变
上升沿	0	0	0	1	0	1
上升沿	0	0	1	0	1	0
上升沿	0	0	1	1	翻转	翻转
x	0	1	x	x	0	1
x	1	0	x	x	1	0
x	1	1	x	x	0	1

S-R触发器（S-R Flip-Flop）



输入					输出	
CLK	Set	Clr	S	R	Q	/Q
上升沿	0	0	0	0	保持不变	保持不变
上升沿	0	0	0	1	0	1
上升沿	0	0	1	0	1	0
上升沿	0	0	1	1	保持不变	保持不变
x	0	1	x	x	0	1
x	1	0	x	x	1	0
x	1	1	x	x	0	1

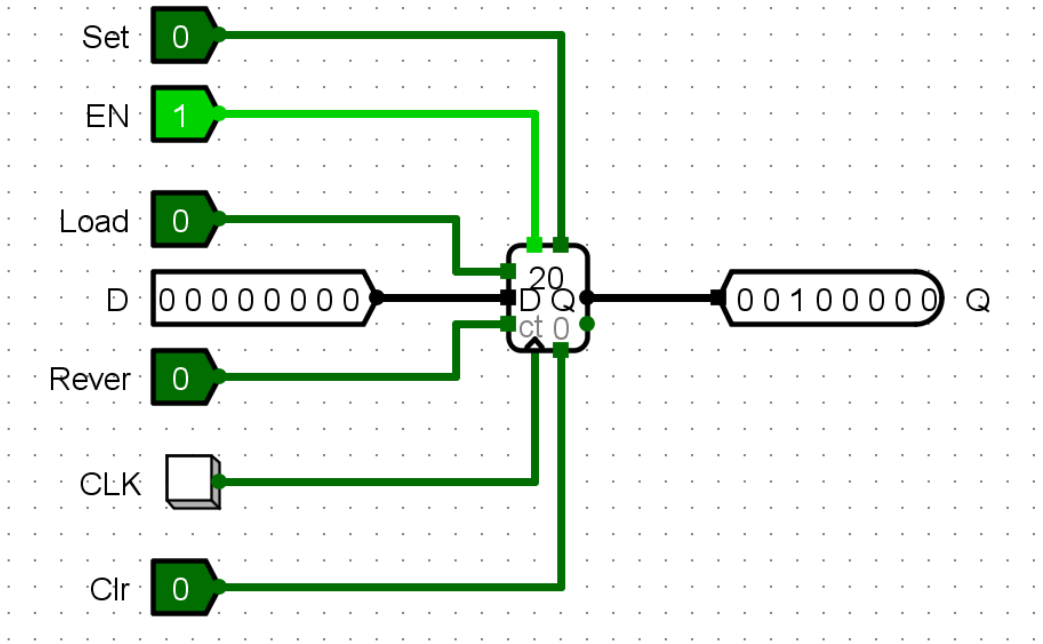
寄存器（Register）



8位寄存器

输入						输出
CLK	CS	EN	Set	Clr	D	Q
x	1	x	x	x	任意	高阻
上升沿	0	1	0	0	任意	D
x	0	0	0	0	任意	保持不变
x	0	x	1	0	任意	全 1
x	0	x	0	1	任意	全 0
x	0	x	1	1	任意	全 0

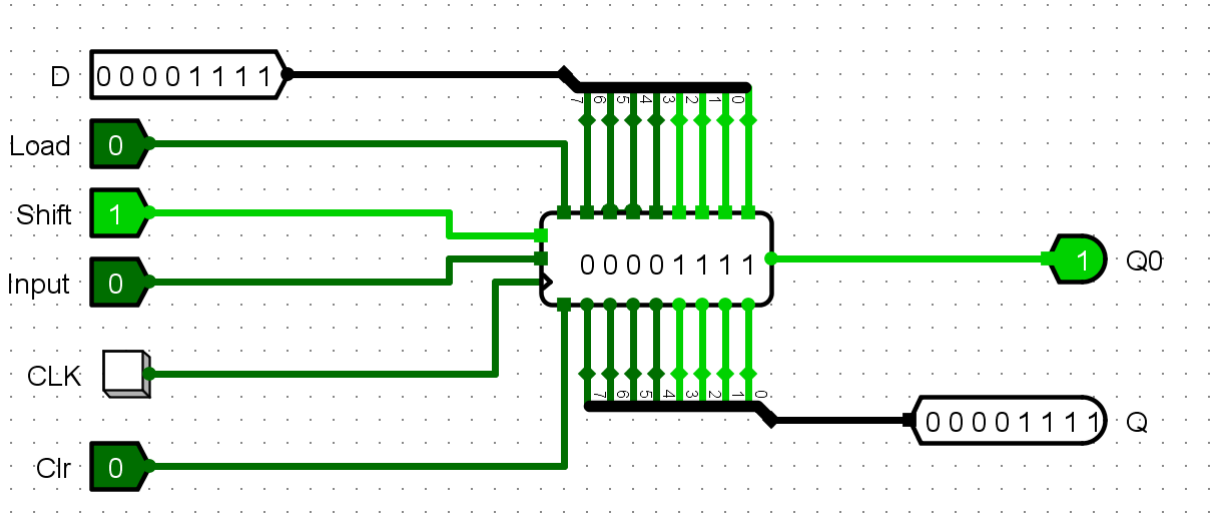
计数器（Counter）



8位计数器

输入							输出
CLK	EN	Load	Rever	Set	Clr	D	Q
上升沿	1	0	0	0	0	任意	计数（加 1）
上升沿	1	0	1	0	0	任意	计数（减 1）
上升沿	1	1	0	0	0	任意	D（装入）
x	x	x	x	1	0	任意	全 1
x	x	x	x	0	1	任意	全 0
x	x	x	x	1	1	任意	全 0

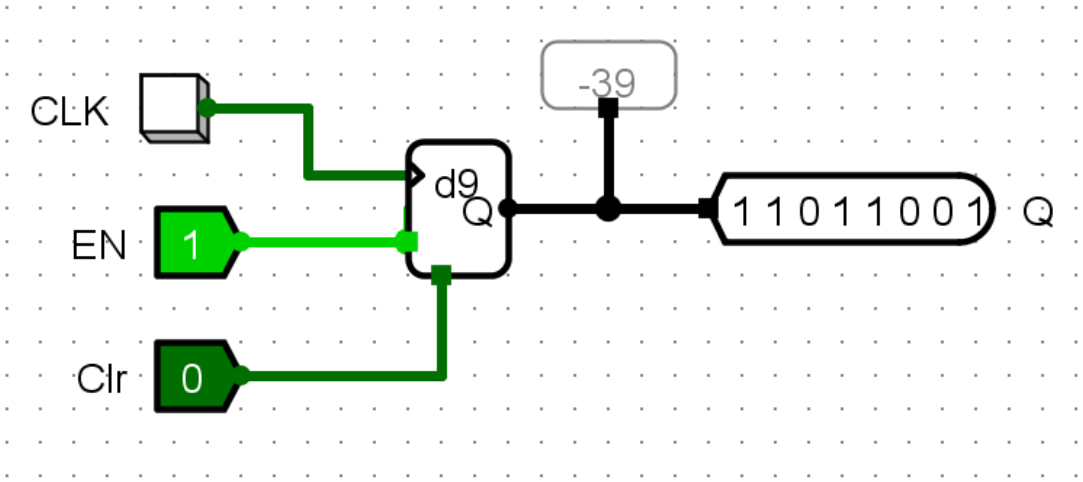
移位寄存器（Shift Register）



输入						输出	
CLK	Load	Shift	Input	Clr	D	Q	Q0
上升沿	1	0	0	0	任意	D（装入）	Q的最低位
上升沿	0	1	0	0	任意	移位（右移）	
上升沿	1	1	0	0	任意	移位（右移）、并将1从左边移入	
x	x	x	x	1	任意	全0	

8位移位寄存器

随机数发生器（Random Generator）

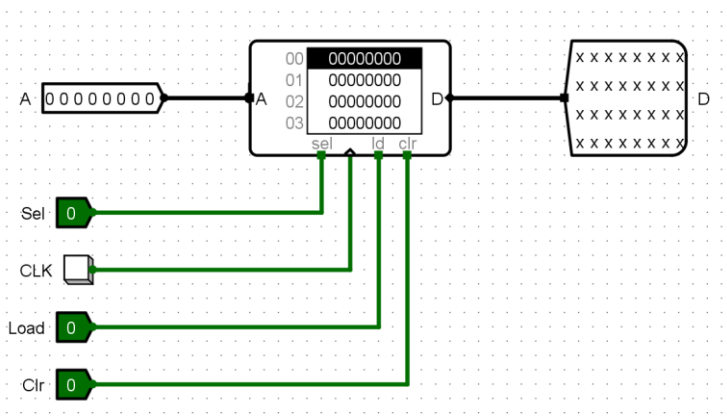


输入			输出
CLK	EN	Clr	Q
上升沿	1	0	产生一个随机数
上升沿	0	0	保持不变

8位随机数发生器

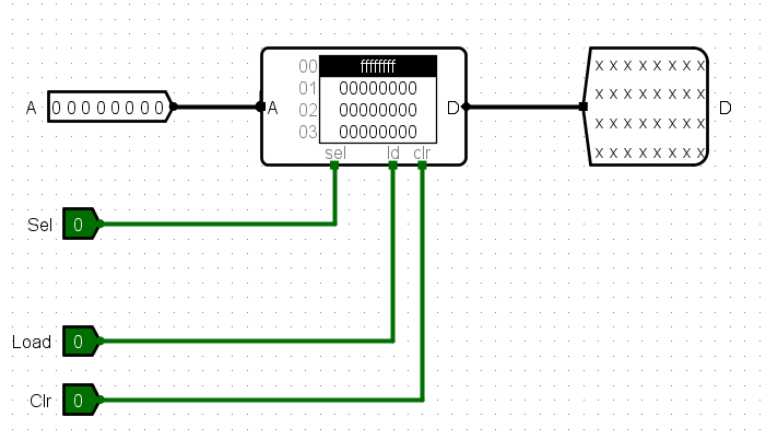
RAM存储器

同步模式



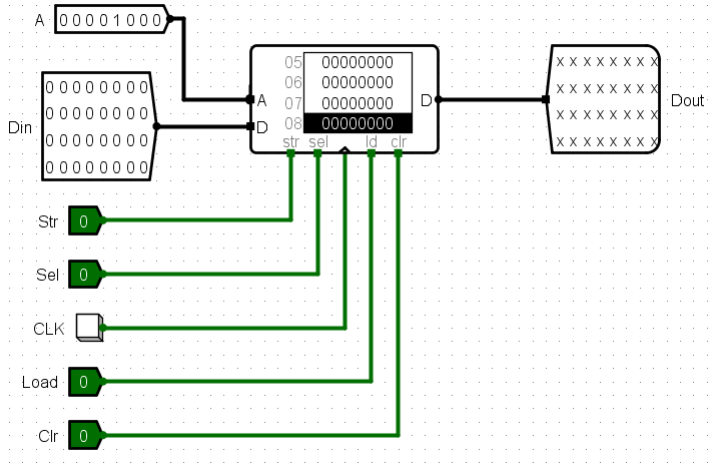
RAM	
Address Bit Width	8
Data Bit Width	32
Label	
Label Font	Dialog Plain 12
Label Color	#000000
Data Interface	One synchronous load/store port
Sel Active On:	Low Level

异步模式



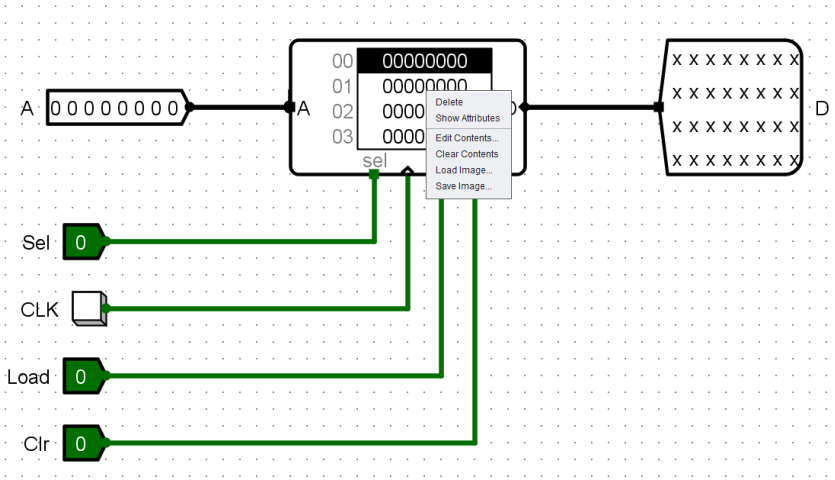
RAM	
Address Bit Width	8
Data Bit Width	32
Label	
Label Font	Dialog Plain 12
Label Color	#000000
Data Interface	One asynchronous load/store port
Sel Active On:	Low Level

分离模式



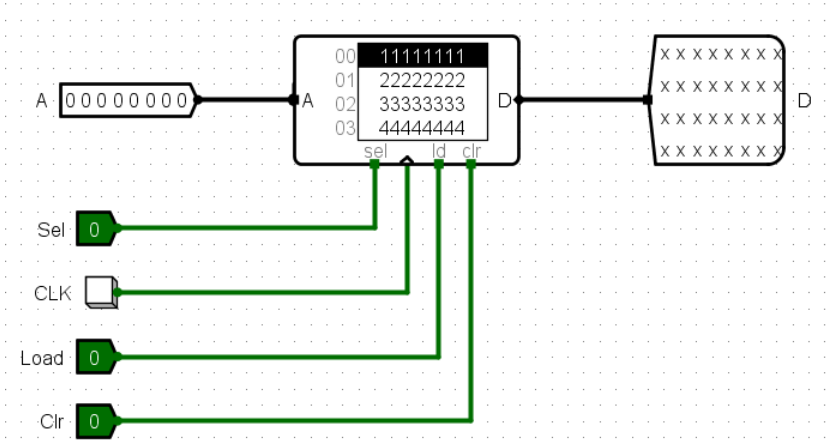
RAM	
Address Bit Width	8
Data Bit Width	32
Label	
Label Font	Dialog Plain 12
Label Color	#000000
Data Interface	Separate load and store ports
Sel Active On:	Low Level

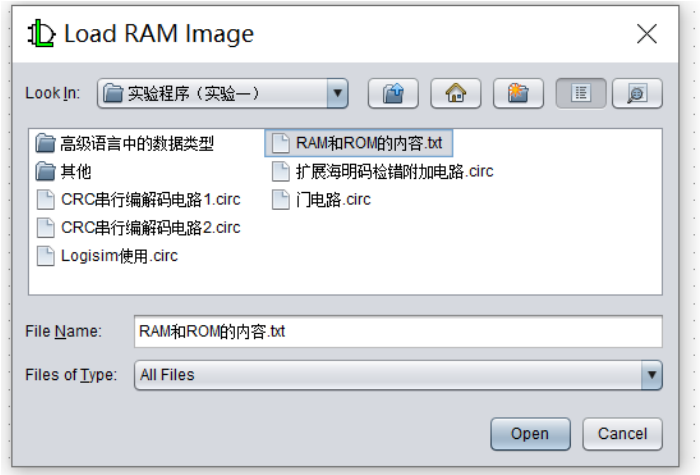
同步模式RAM存储器的操作

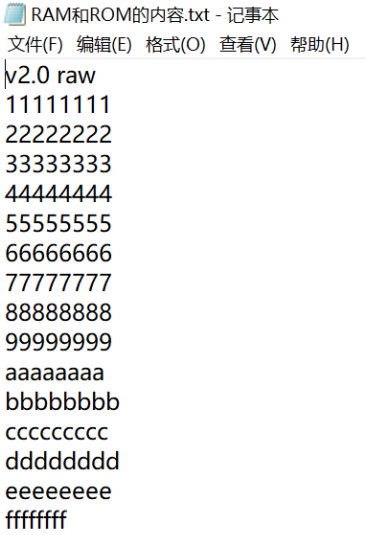


装入数据

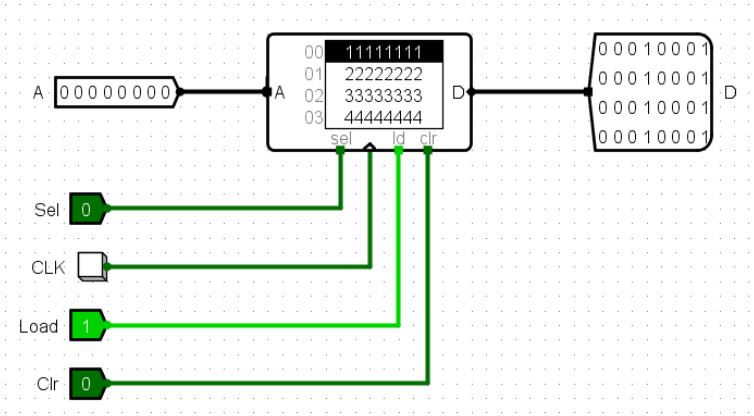
Sel=0、Load=0、Clr=0；按鼠标右键，选择“Load Image”，将文本文件中的内容装入存储器中





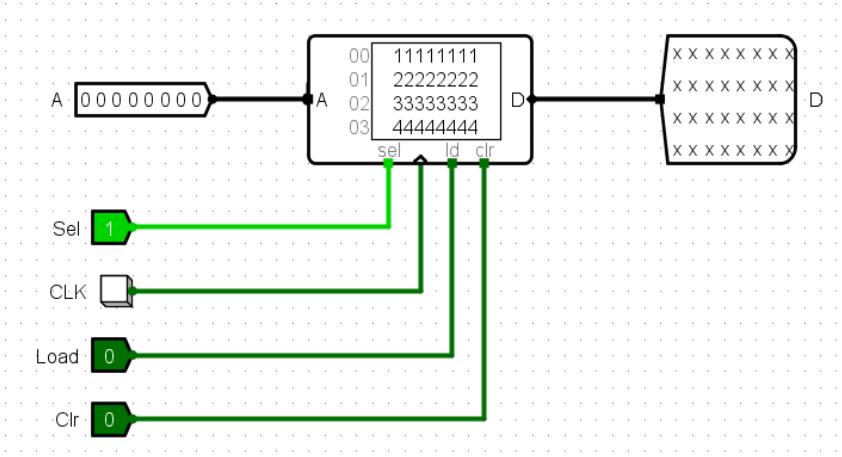


同步模式RAM存储器的操作



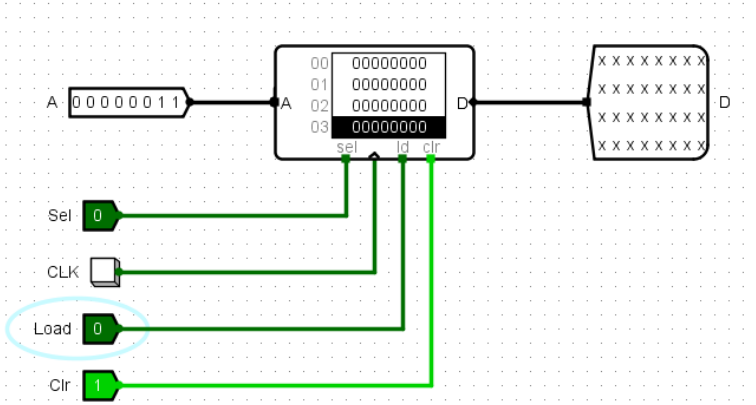
读出数据

Sel=0、Load=1、Clr=0；给定地址A，则对应的内容输出到D



输出高阻

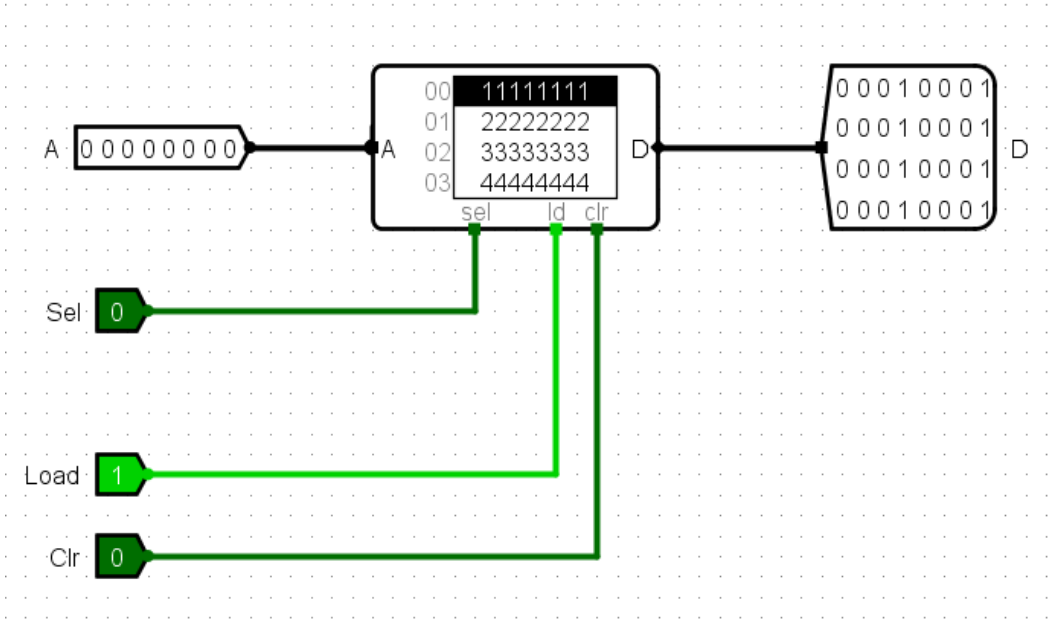
Sel=1、Load=0、Clr=0时，输出为高阻



内容清零

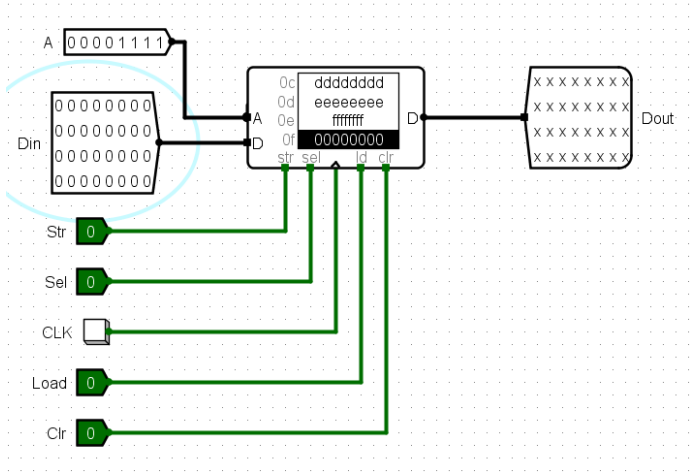
Sel=0、Load=0、Clr=1时，存储器中的内容清零

异步模式RAM存储器的操作



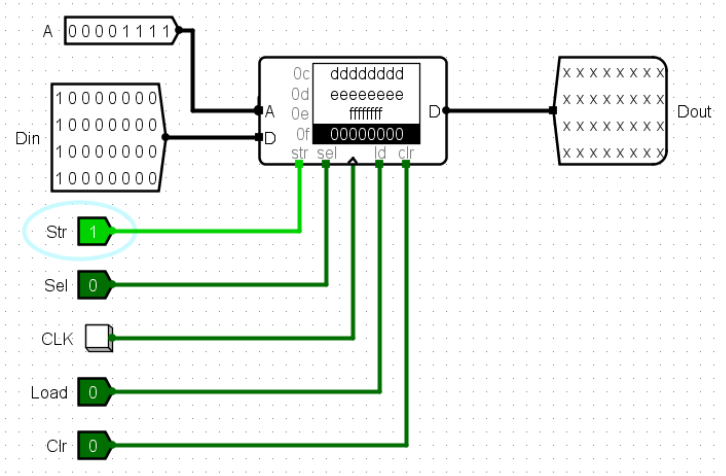
除了没有时钟信号CLK外，其它与同步模式相同

分离模式RAM存储器的操作



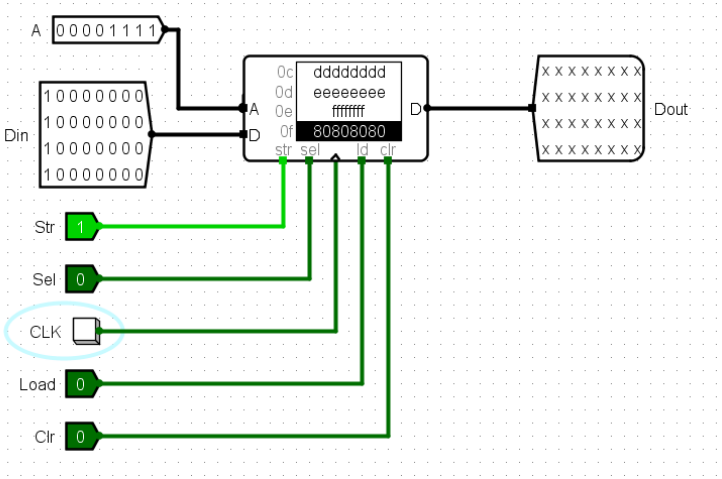
装入数据

Str=0、Sel=0、Load=0、Clr=0；按鼠标右键，选择“Load Image”，将文本文件中的内容装入存储器中



写入数据（1）

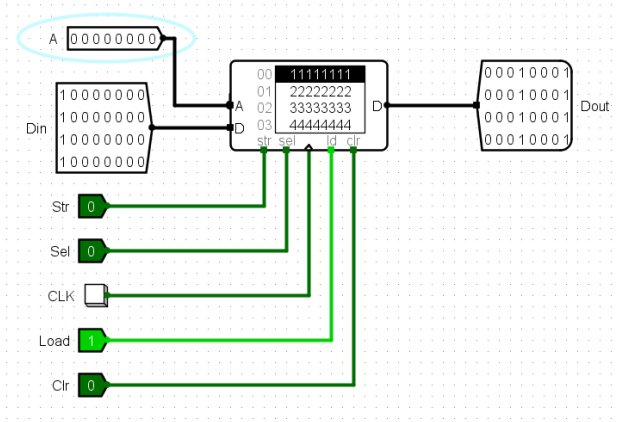
Str=1、Sel=0、Load=0、Clr=0；A=00001111、Din=80808080H



写入数据（2）

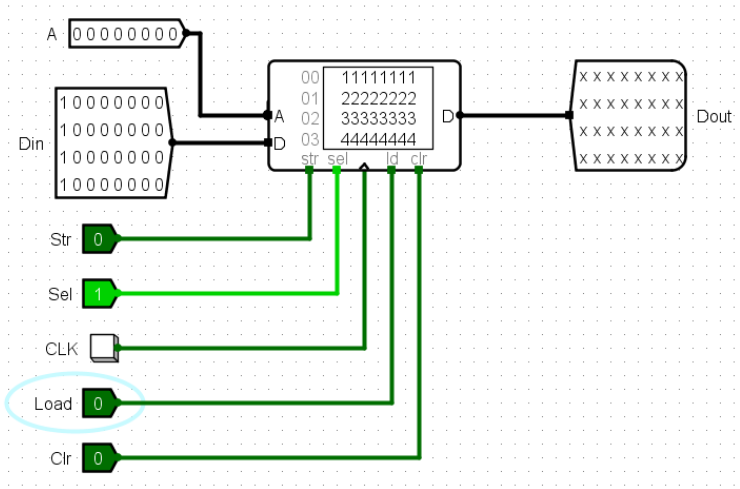
按CLK键，则将Din写入到00001111存储单元

分离模式RAM存储器的操作



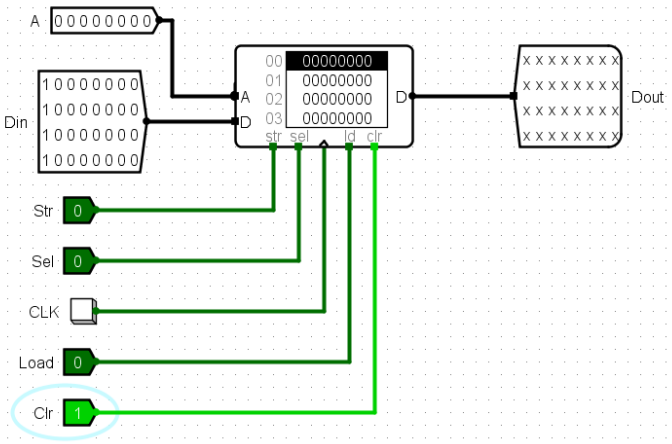
读出数据

Str=0、Sel=0、Load=1、Clr=0；给定地址A，则对应的内容输出到D中



输出高阻

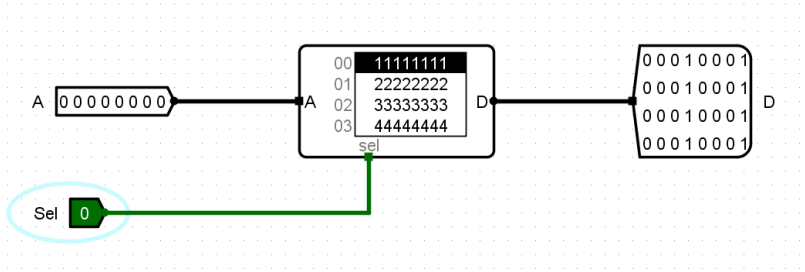
Str=0、Sel=1、Load=0、Clr=0；A=00001111、Din=80808080H



内容清零

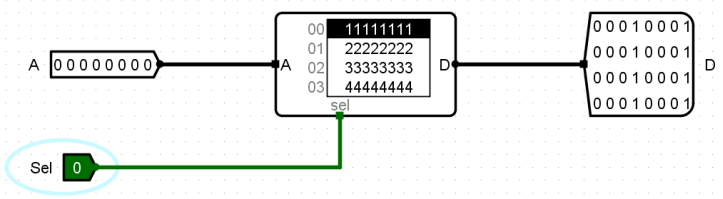
Str=0、Sel=0、Load=0、Clr=1；A=00001111、Din=80808080H

ROM存储器



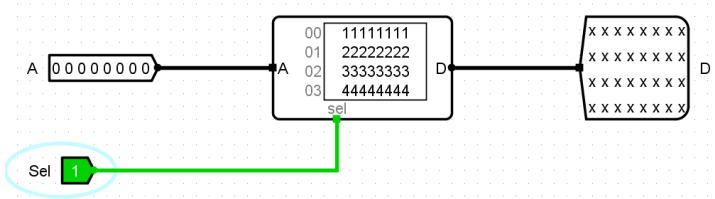
装入数据

Sel=0；按鼠标右键，选择“Load Image”，将文本文件中的内容装入存储器中



读出数据

Sel=0；给定地址A，则对应的内容输出到D中

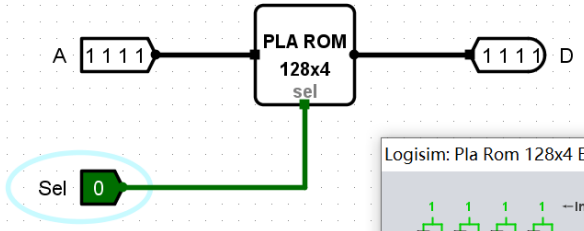


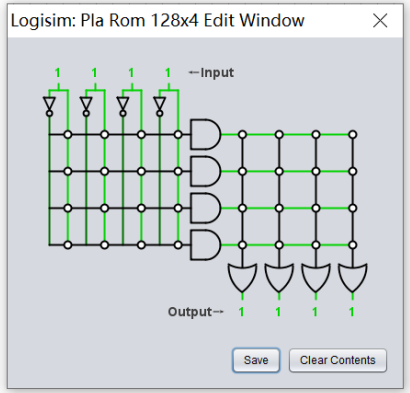
输出高阻

Sel=1时，输出为高阻

PLA ROM

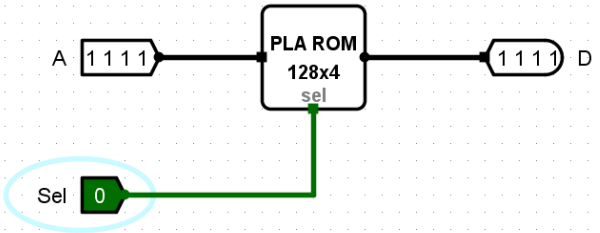
Programmable Logic Arrays：可编程逻辑阵列





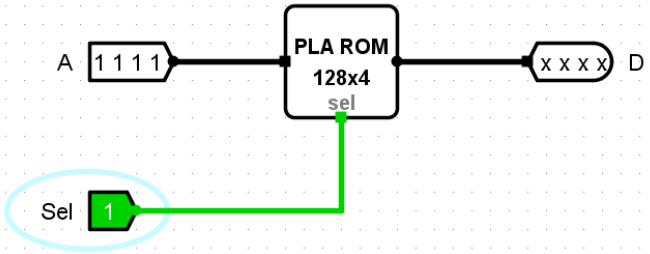
编程（设置内容）

Sel=0；按鼠标右键，编辑框，改变输入、输出矩阵的交叉点，即可完成编程（设置16个单元的内容）



读出数据

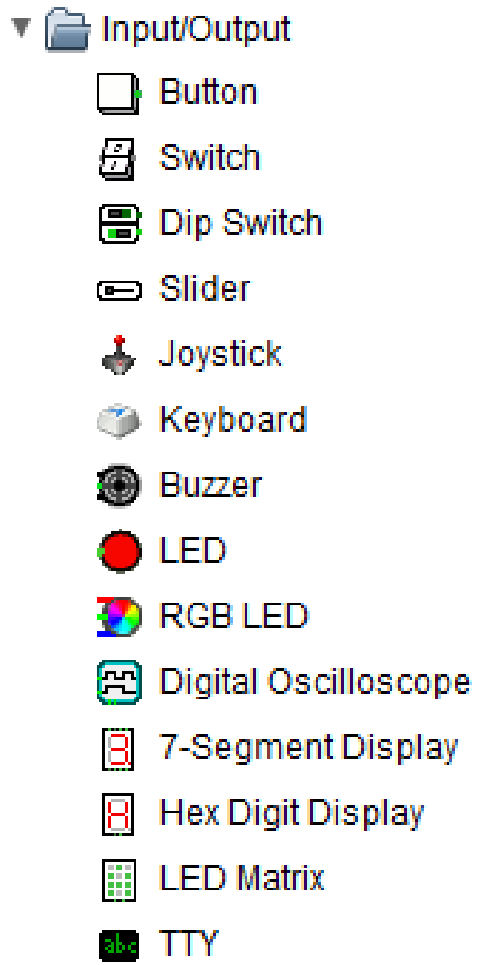
Sel=0；给定地址A，则对应的内容输出到D中



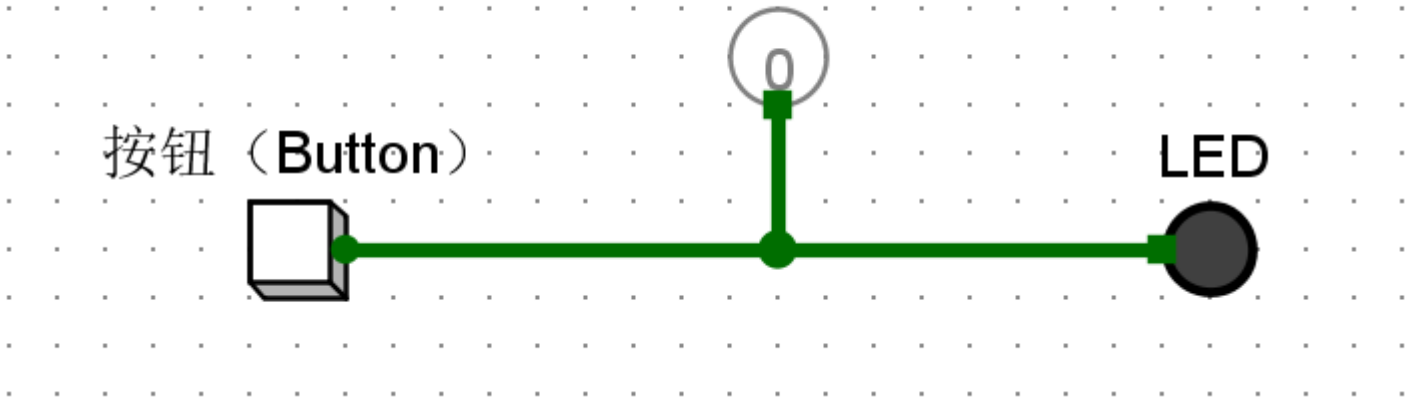
输出高阻

Sel=1时，输出为高阻

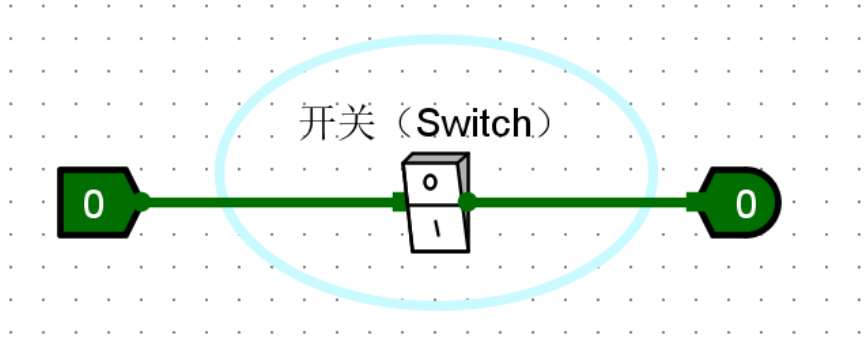
（七）输入/输出部件（Input/Output）

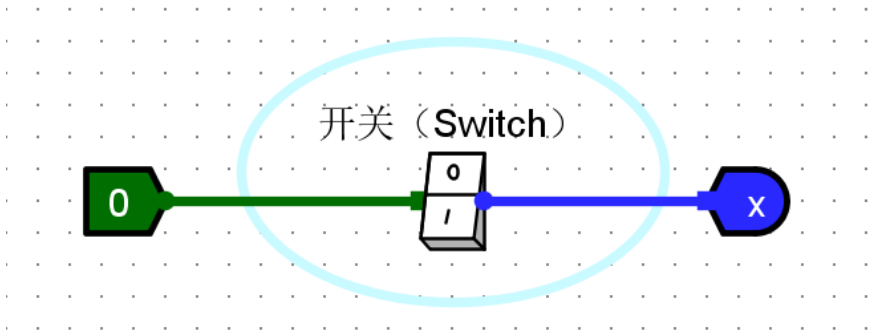


按钮（Button）

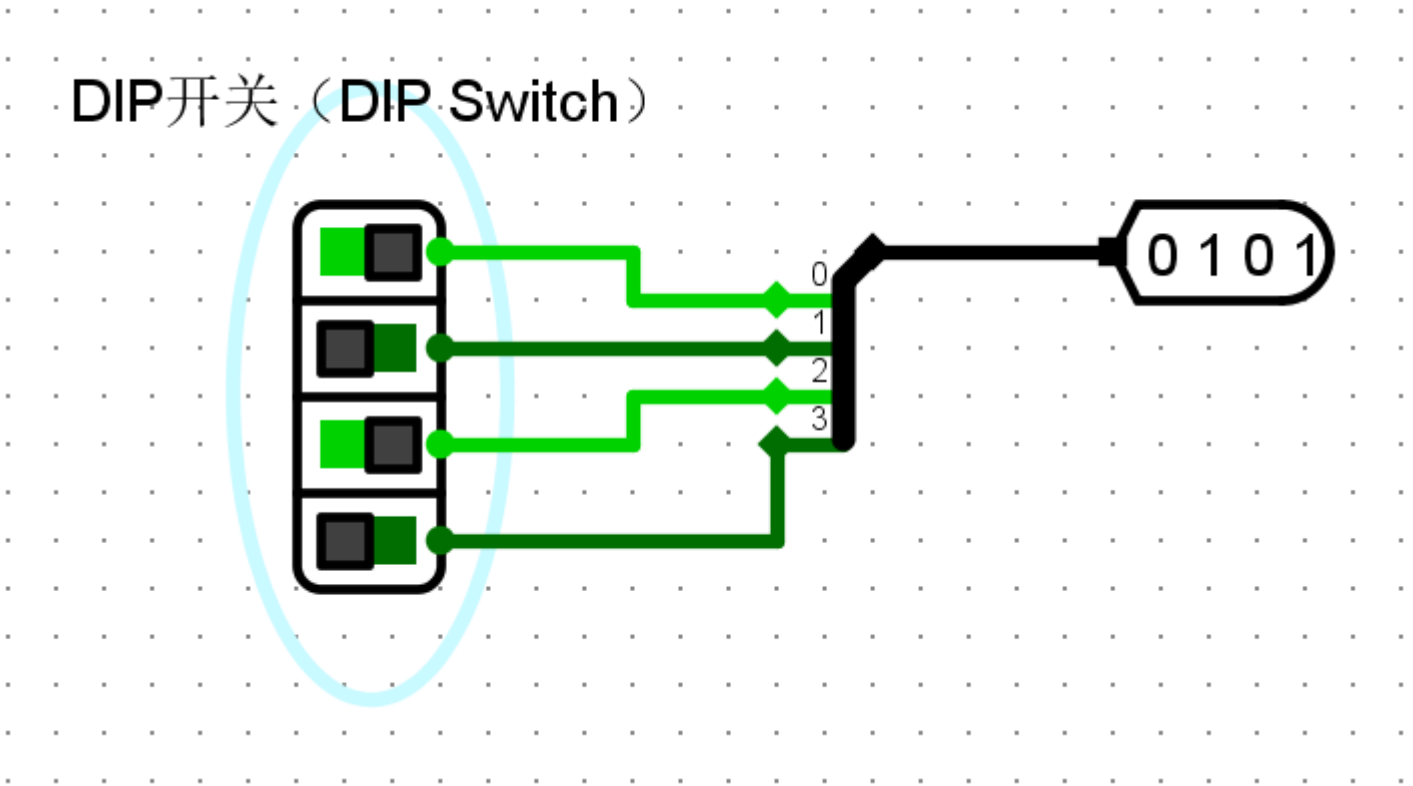


开关（Switch）

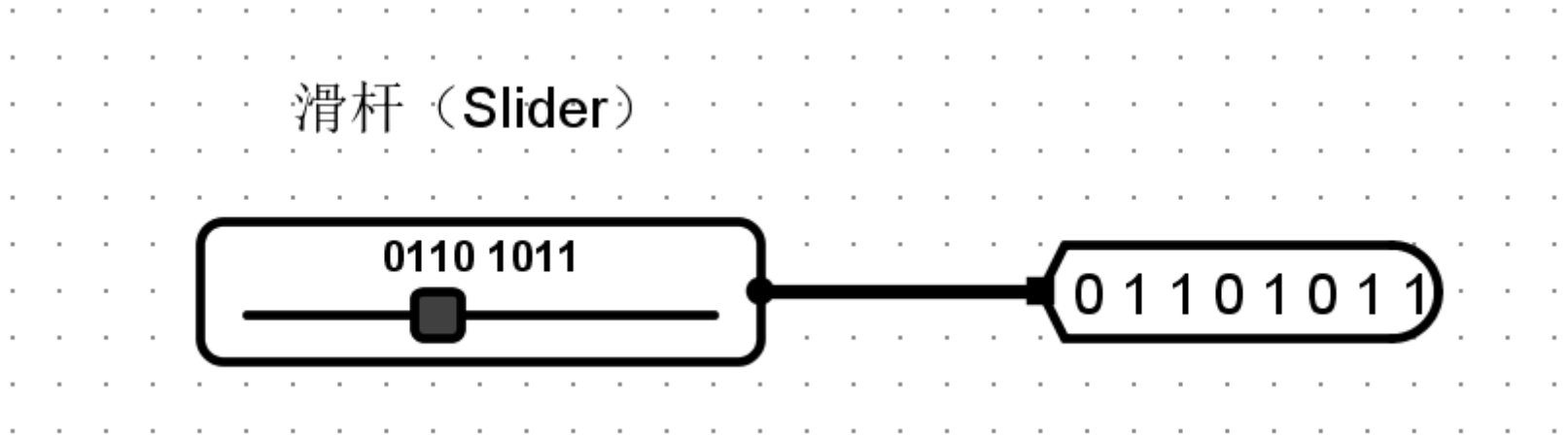




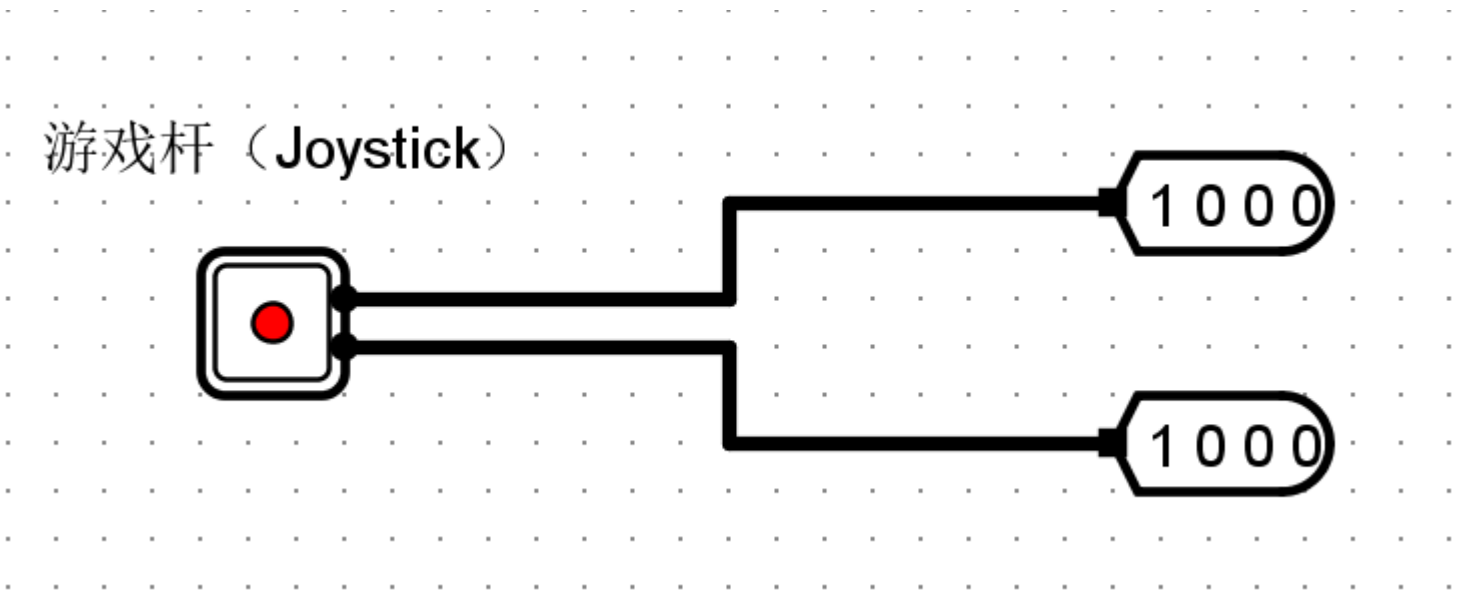
DIP开关（DIP Switch）



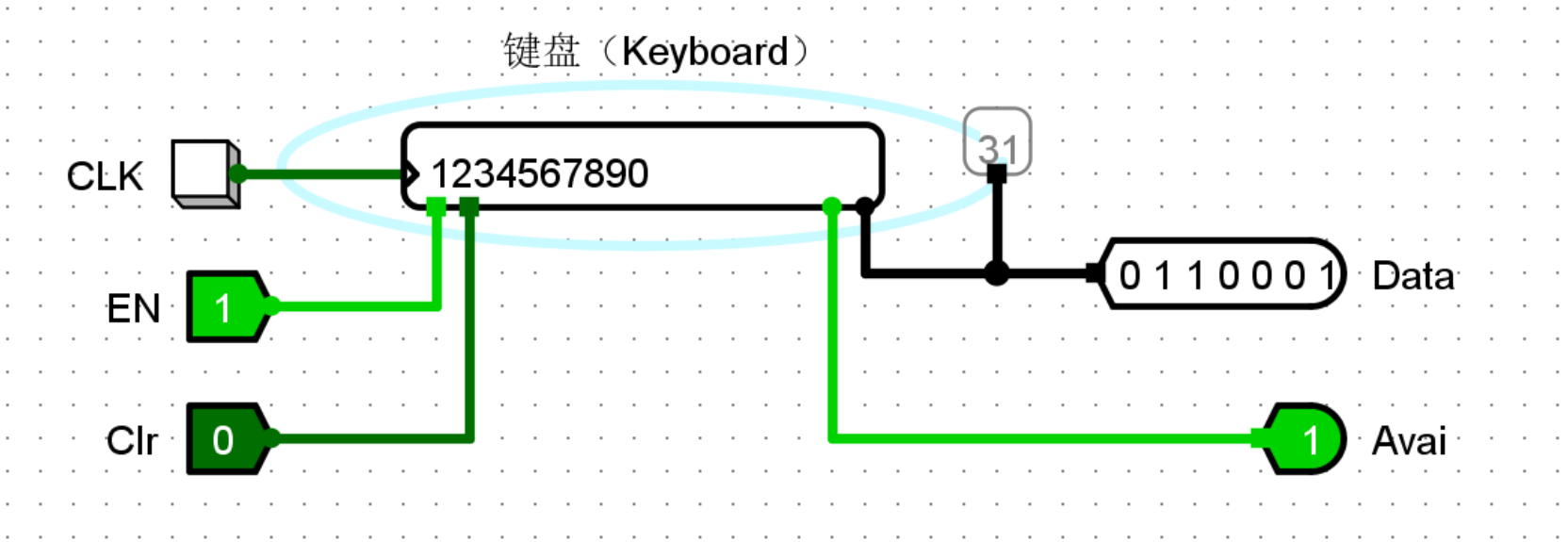
滑杆（Slider）



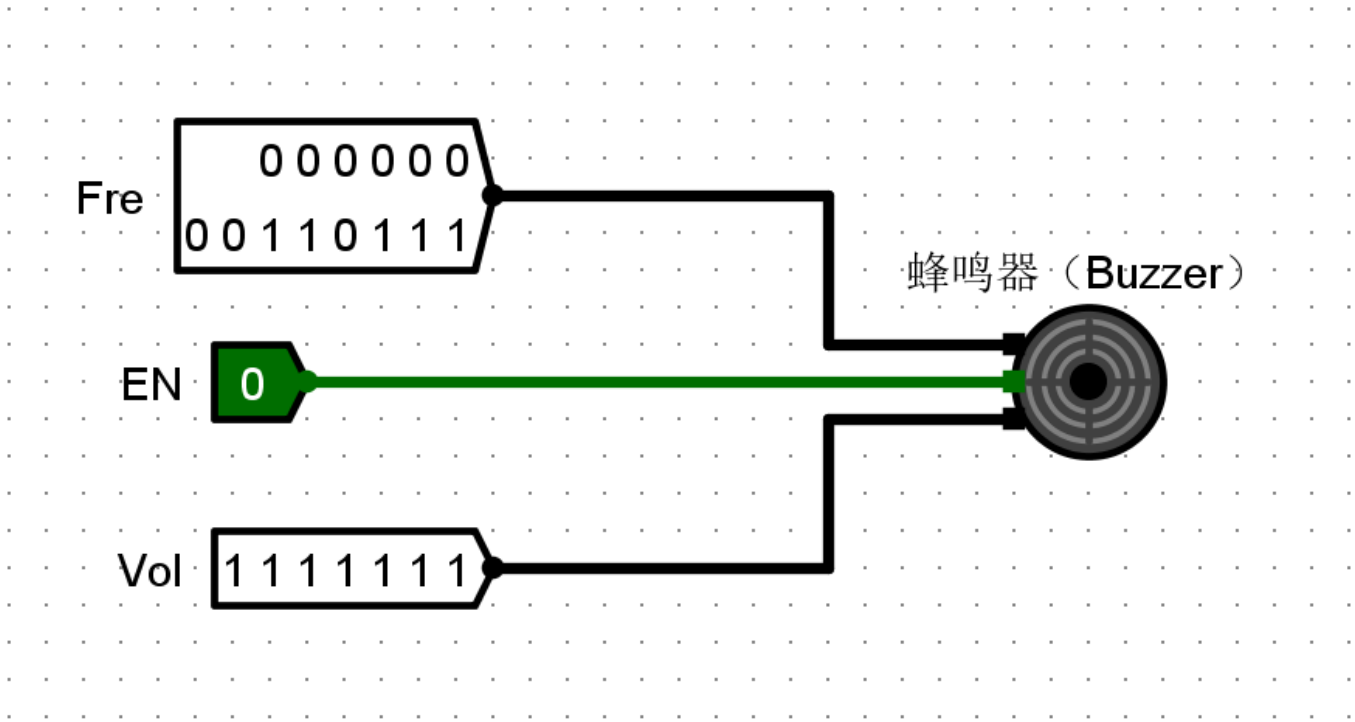
游戏杆（Joystick）



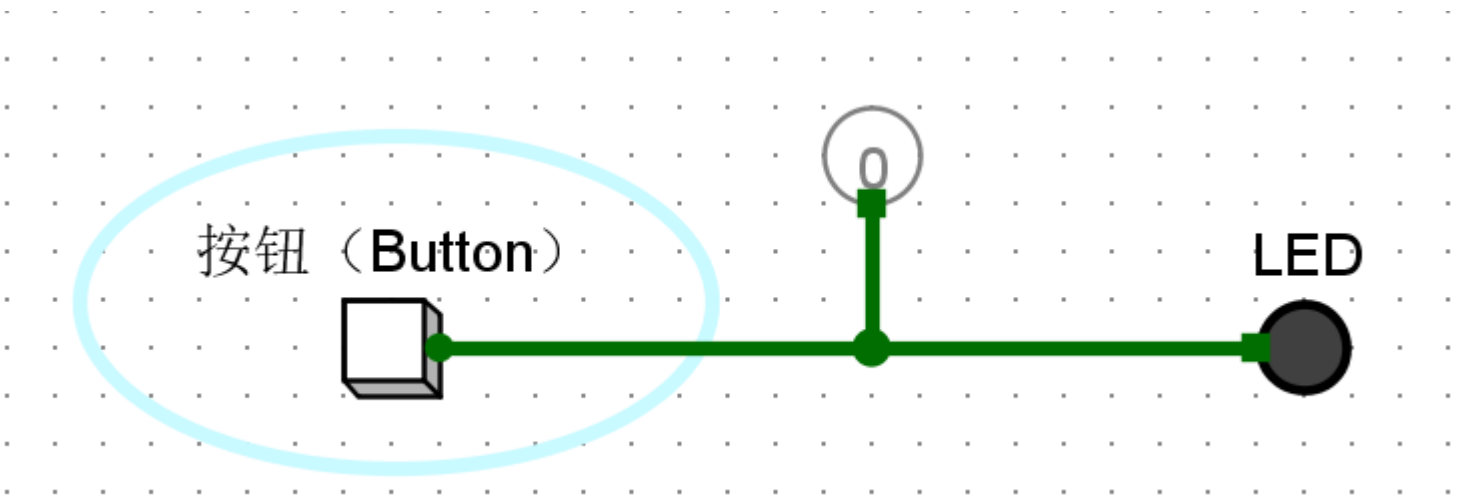
键盘（Keyboard）



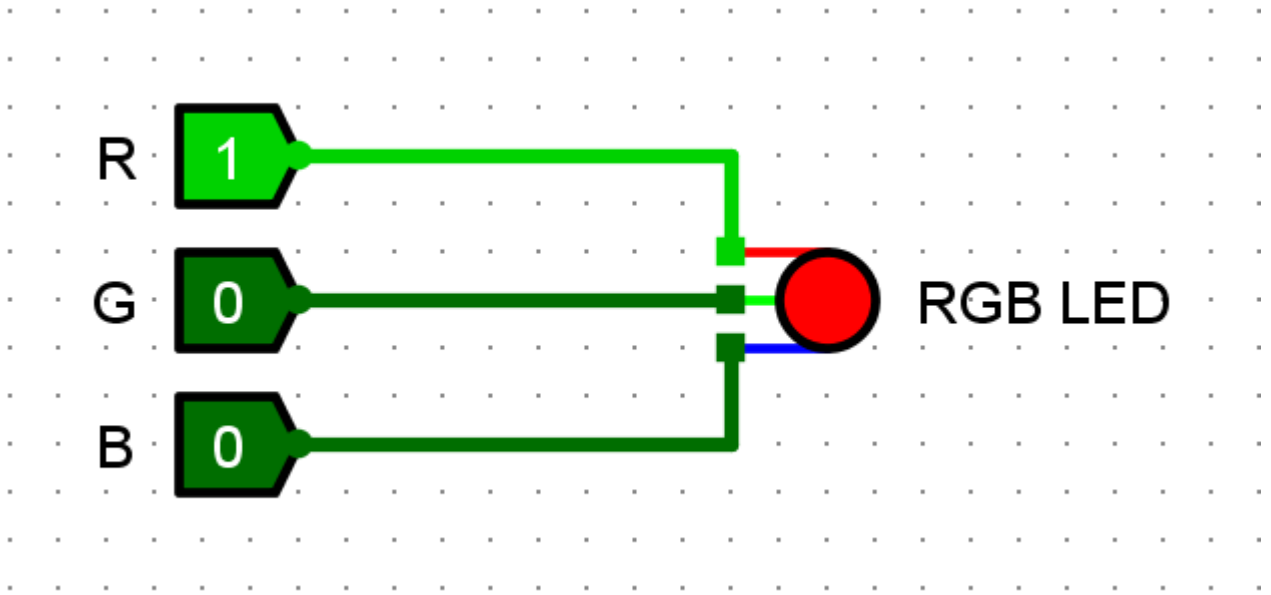
蜂鸣器（Buzzer）



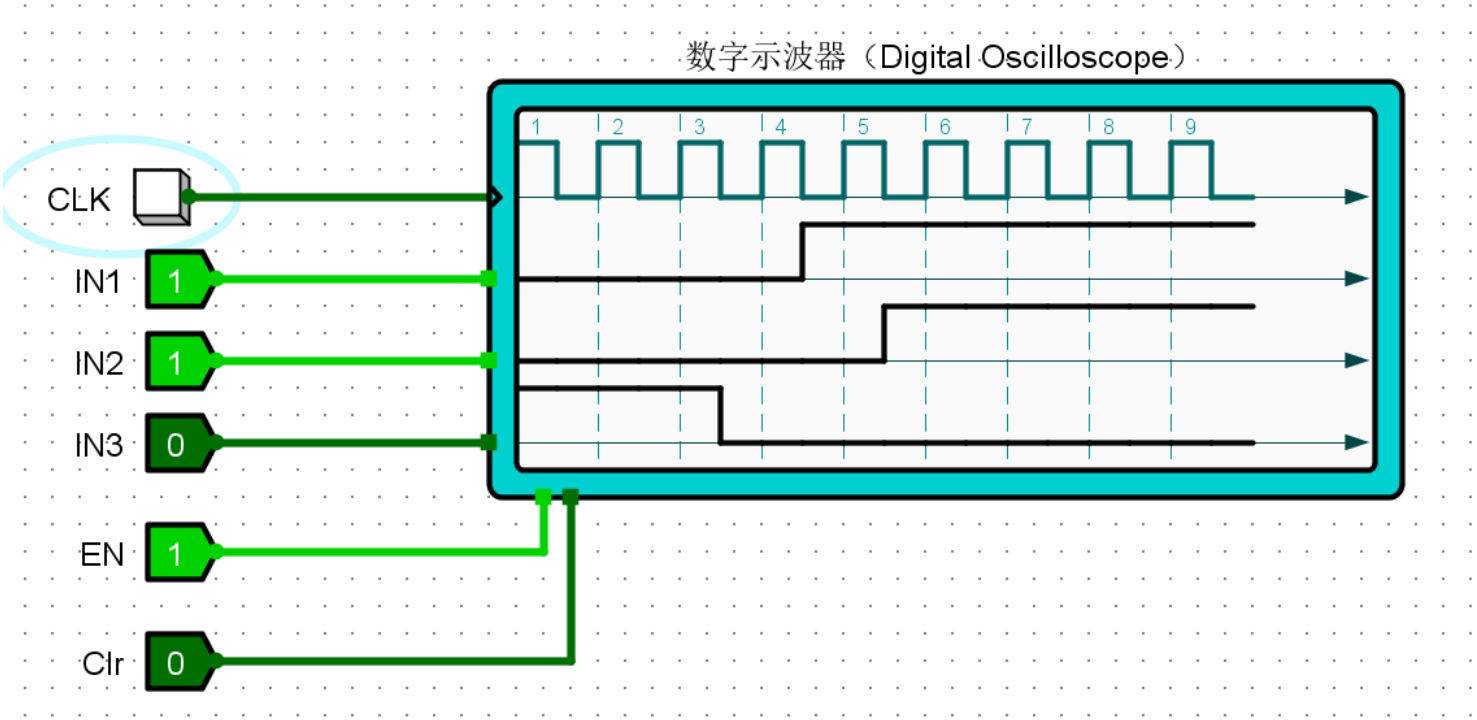
LED灯



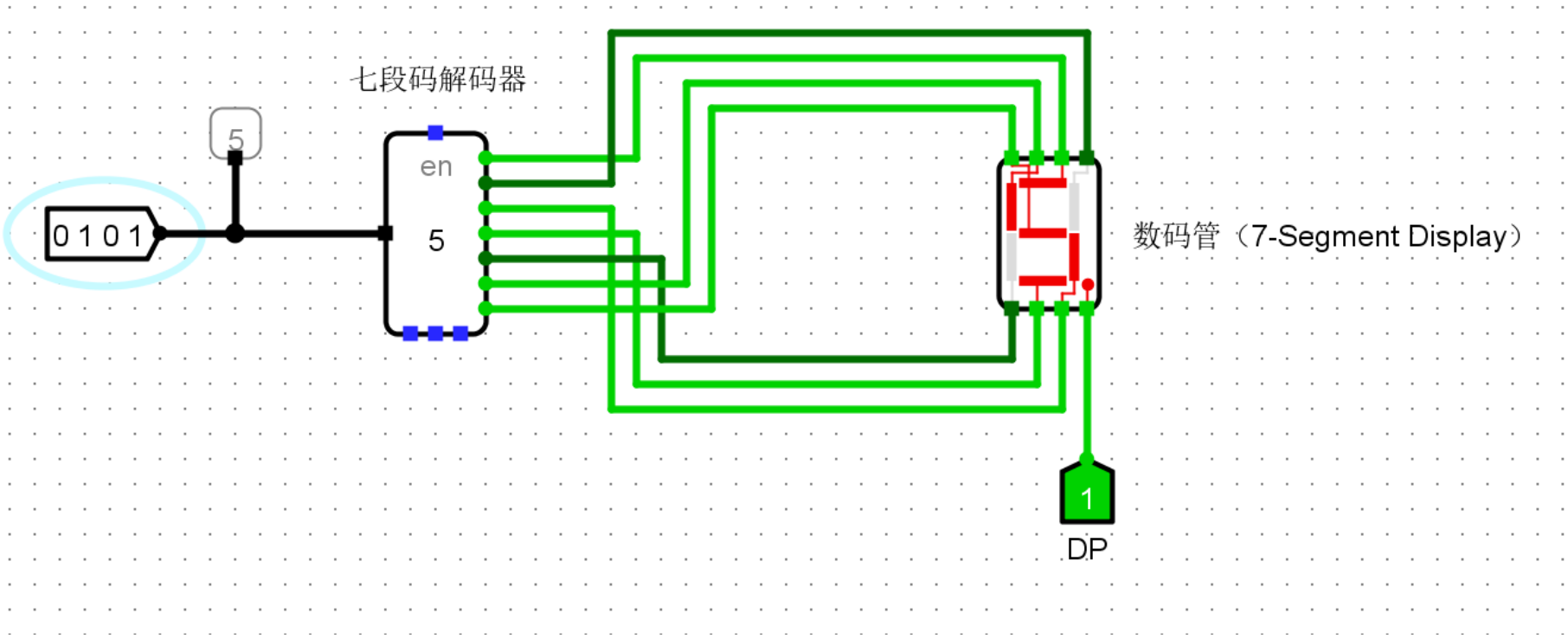
三色灯（RGB LED）



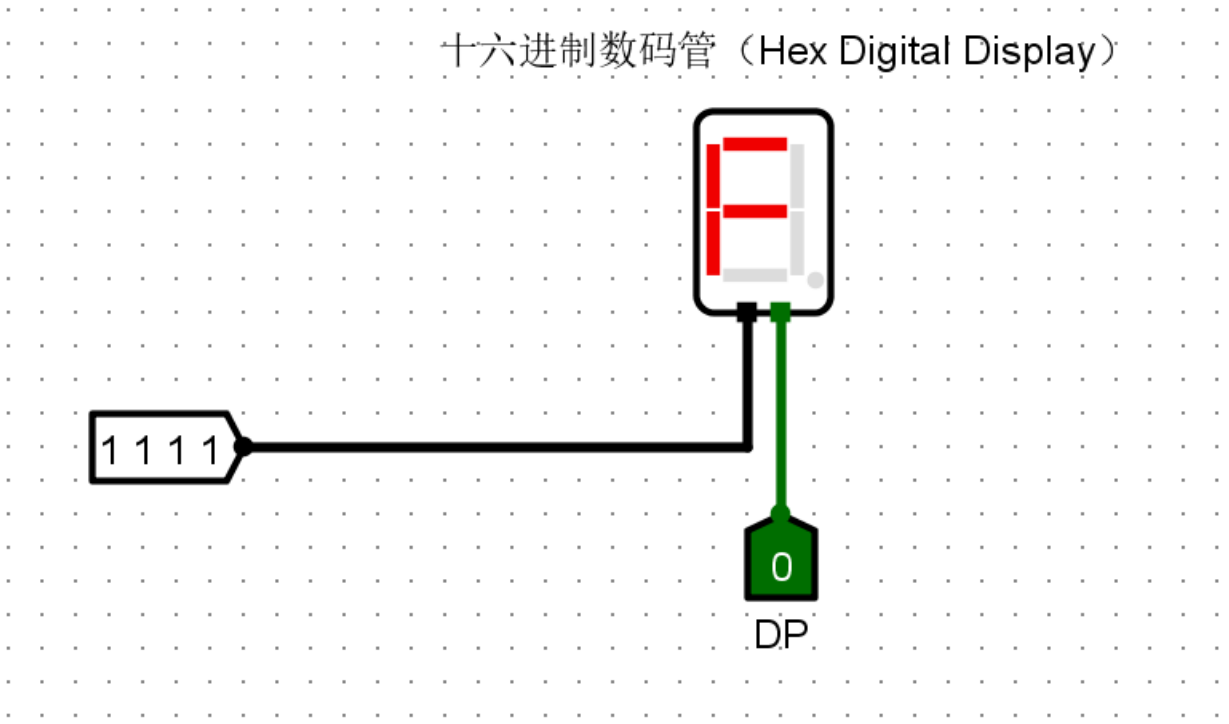
数字示波器（Digital Oscilloscope）



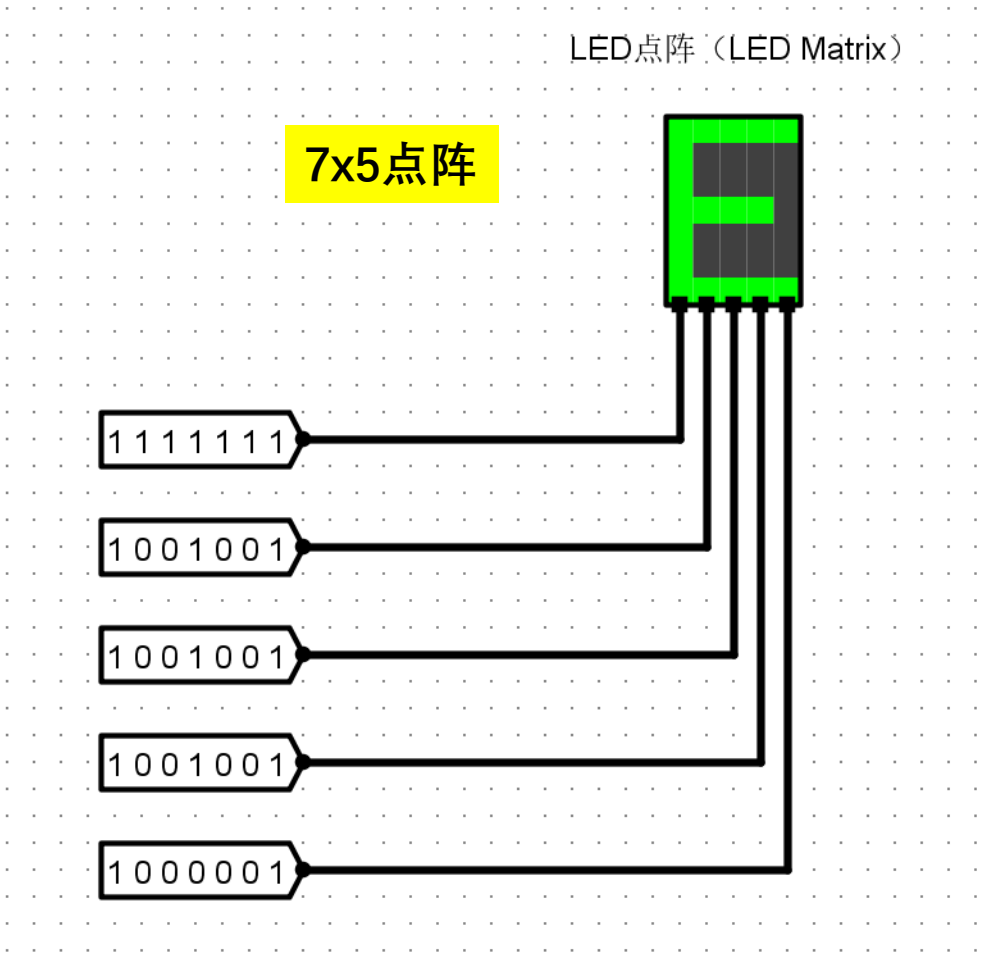
数码管（7-Segment Display）

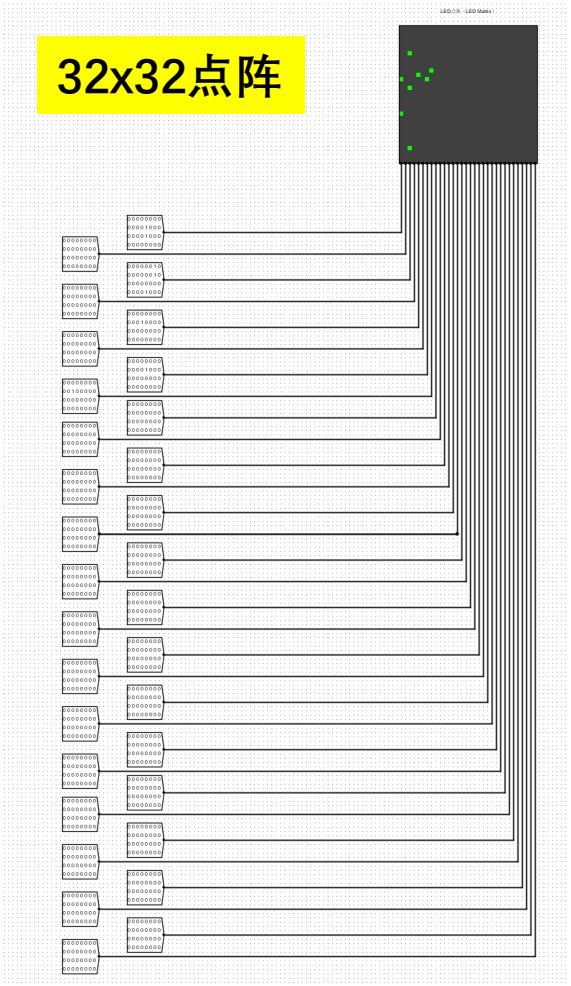


十六进制数码管（Hex Digital Display）

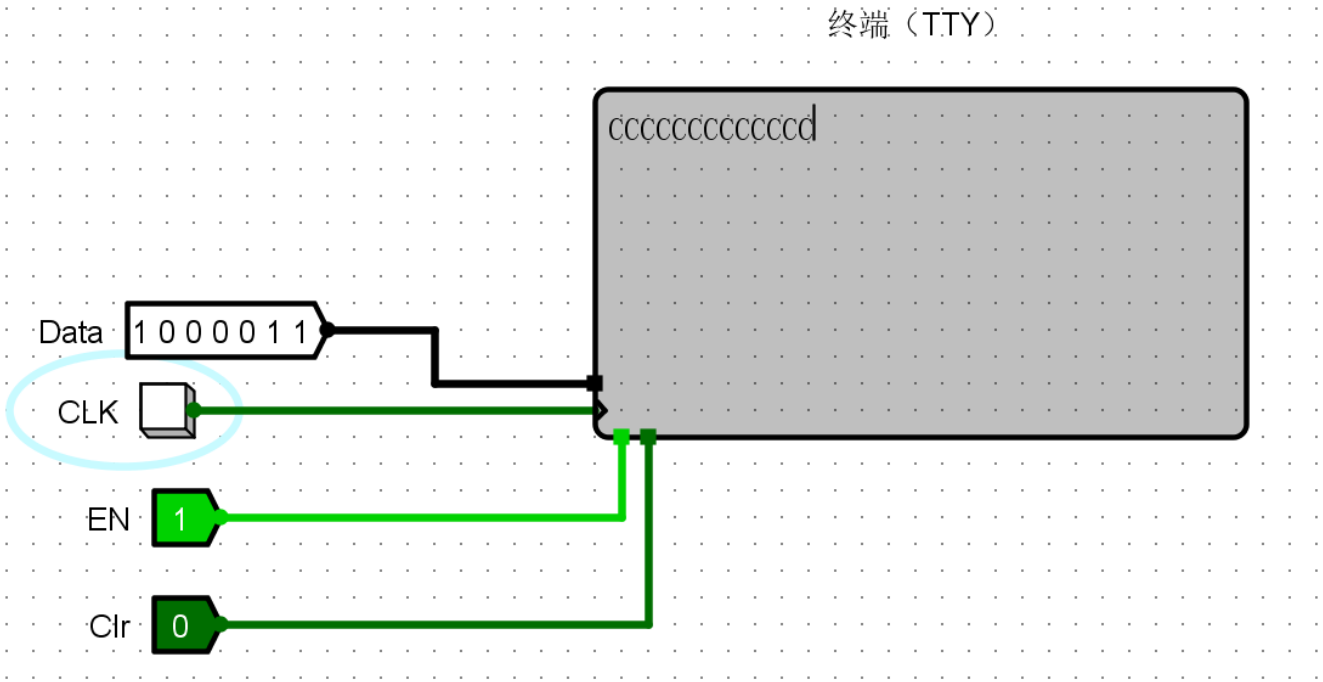


LED点阵（LED Matrix）

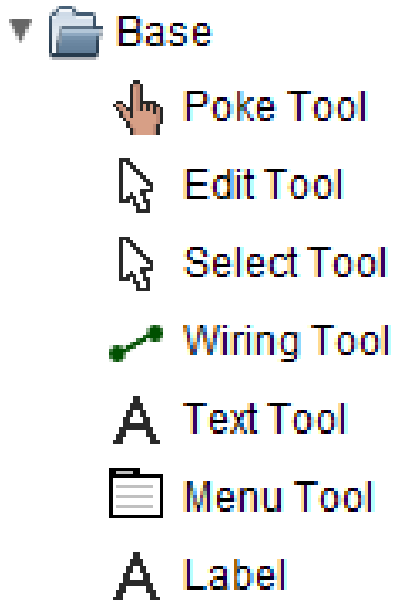




终端（TTY）

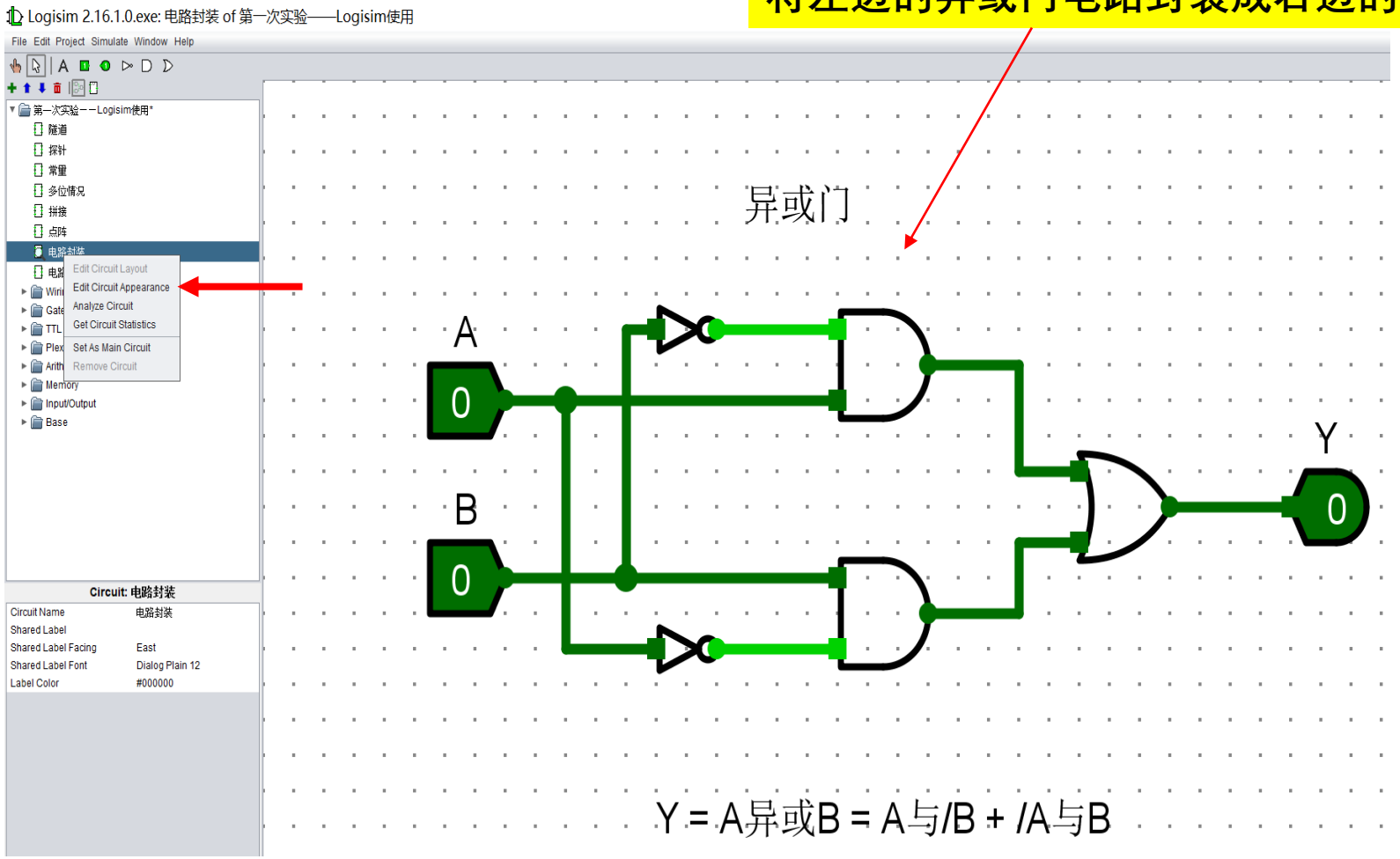


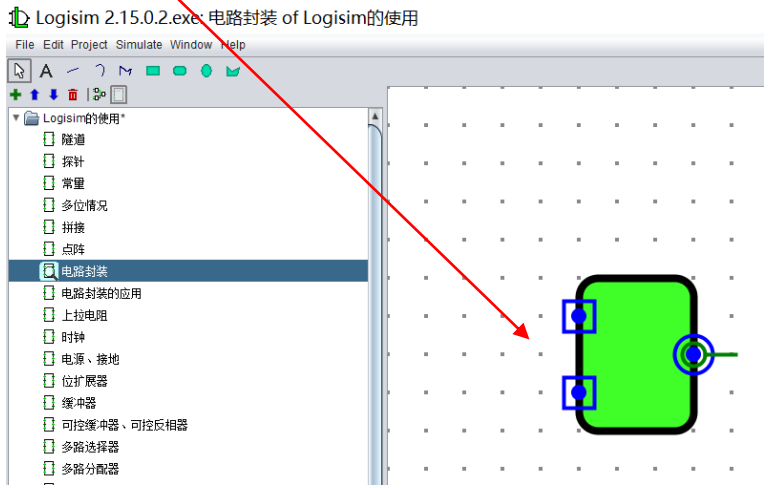
（八）基本操作（Base）


Base
Poke Tool
Edit Tool
Select Tool
Wiring Tool
Text Tool
Menu Tool
Label

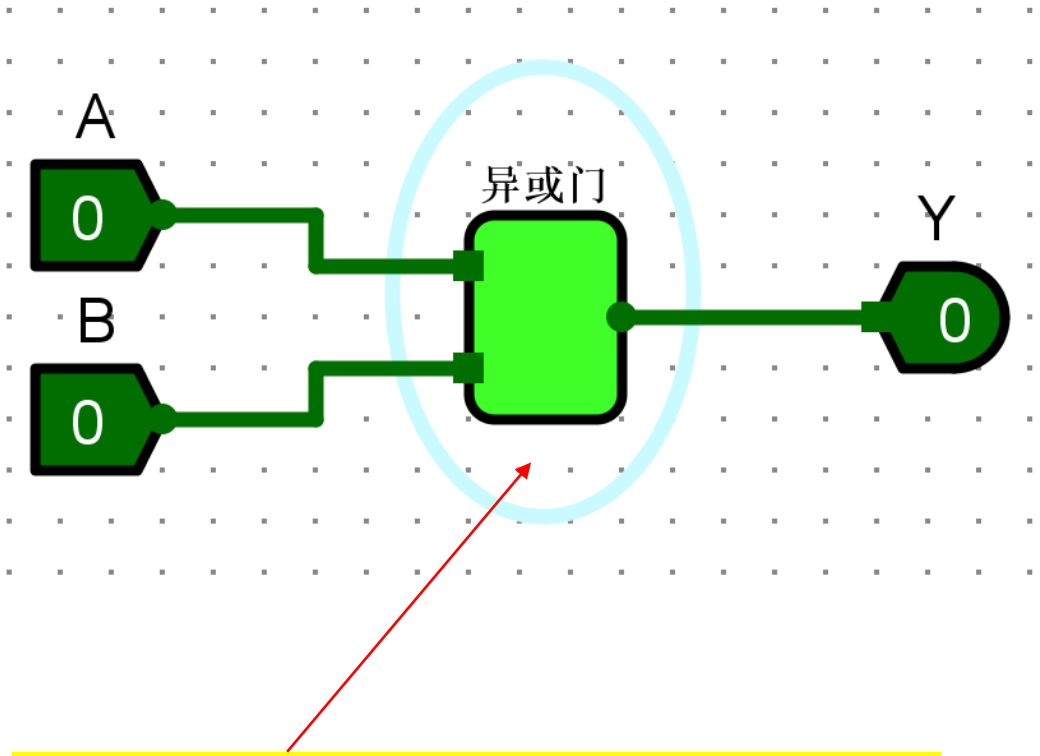
（九）电路封装

将左边的异或门电路封装成右边的电路





电路封装的应用



这里的异或门电路就是上面封装好的电路

Thanks