

《计算机组成原理实验》

(第二次)

厦门大学信息学院软件工程系 曾文华

2023年3月23日

目录

- 一、桶型移位器
- 二、串行加法器
 - 1、8位串行加法器
 - 2、8位串行可控加减法器
- 三、先行进位加法器
 - 1、4位先行进位电路
 - 2、4位快速加法器
 - 3、16位快速加法器（组内并行、组间串行）
 - 4、16位快速加法器（组内并行、组间并行）（设计实验）
 - 5、32位快速加法器（组内并行、组间串行）（设计实验）
- 四、算术逻辑单元（ALU）
- 五、阵列乘法器
 - 1、5位无符号阵列乘法器（斜向）
 - 2、5位无符号阵列乘法器（横向）
 - 3、6位原码阵列乘法器（设计实验）
 - 4、6位补码阵列乘法器（设计实验）
- 六、原码和补码一位乘法器
 - 1、8位无符号一位乘法器
 - 2、8位原码一位乘法器（设计实验）
 - 3、8位补码一位乘法器
 - 4、8位补码一位乘法器（采用8位无符号一位乘法器实现）（挑战性实验）

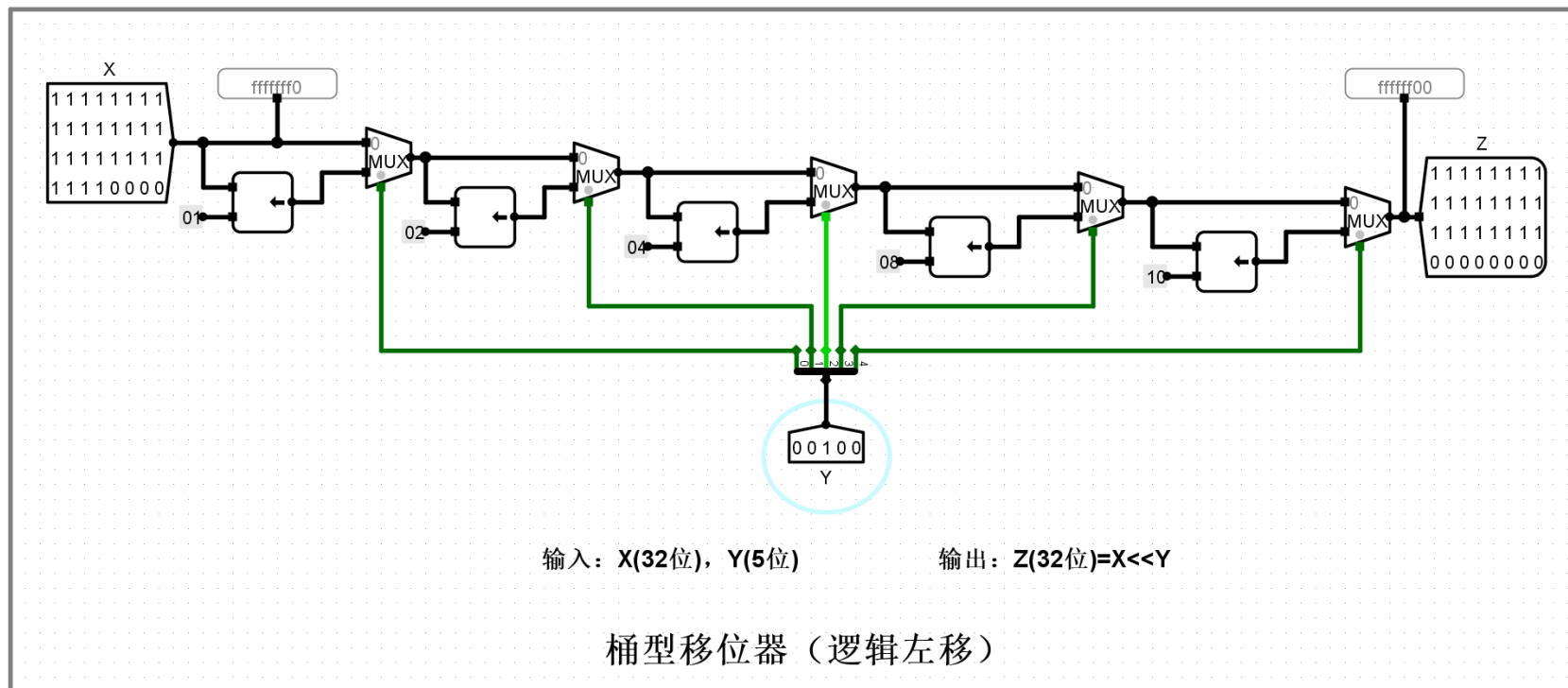
一、桶型移位器

对应教材3.1.3小节（图3.1）

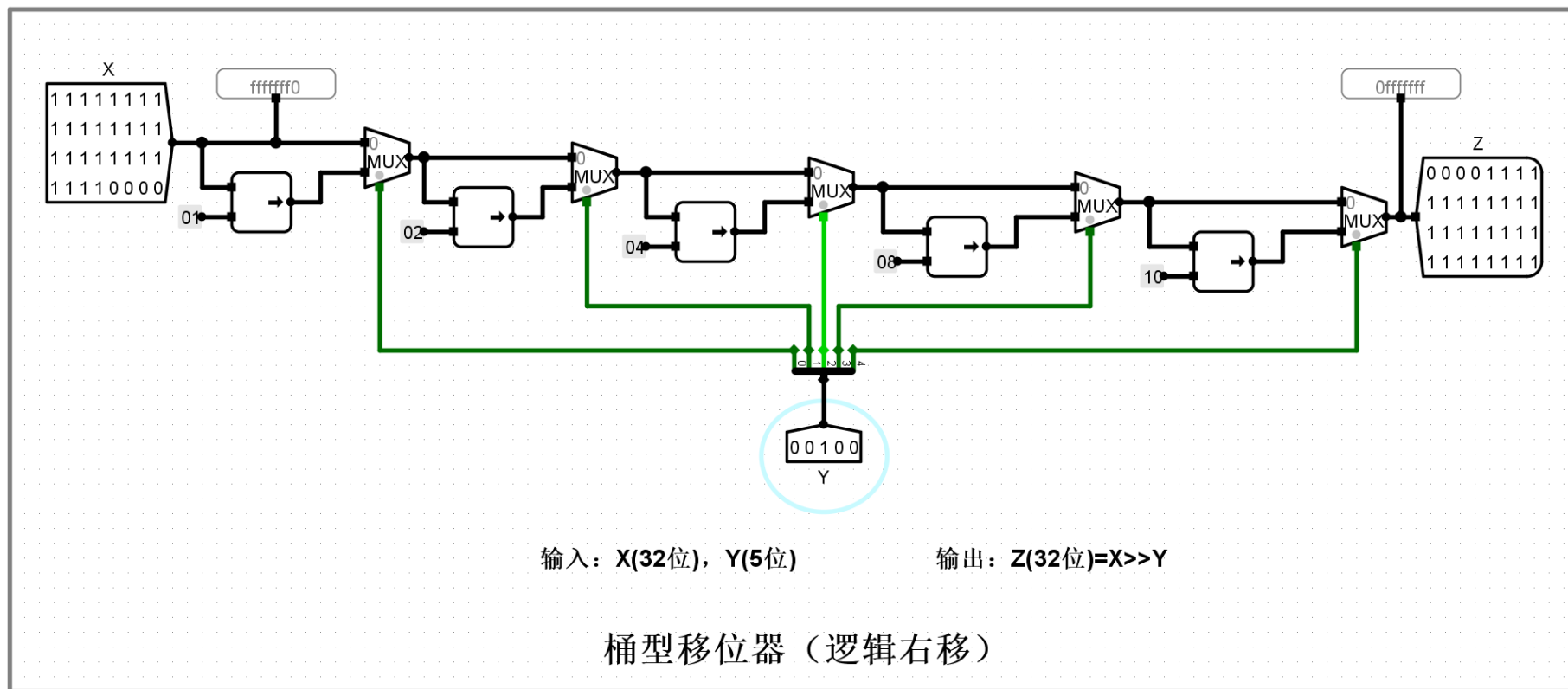
参考设计文件：桶型移位器.circ（请从FTP下载）

一、桶型移位器

- 1、逻辑左移： $Z = X \ll Y$
 - 请验证： $X = \text{FFFFFFF0}$, $Y = 00100$, $Z = \text{FFFFFFF00}$

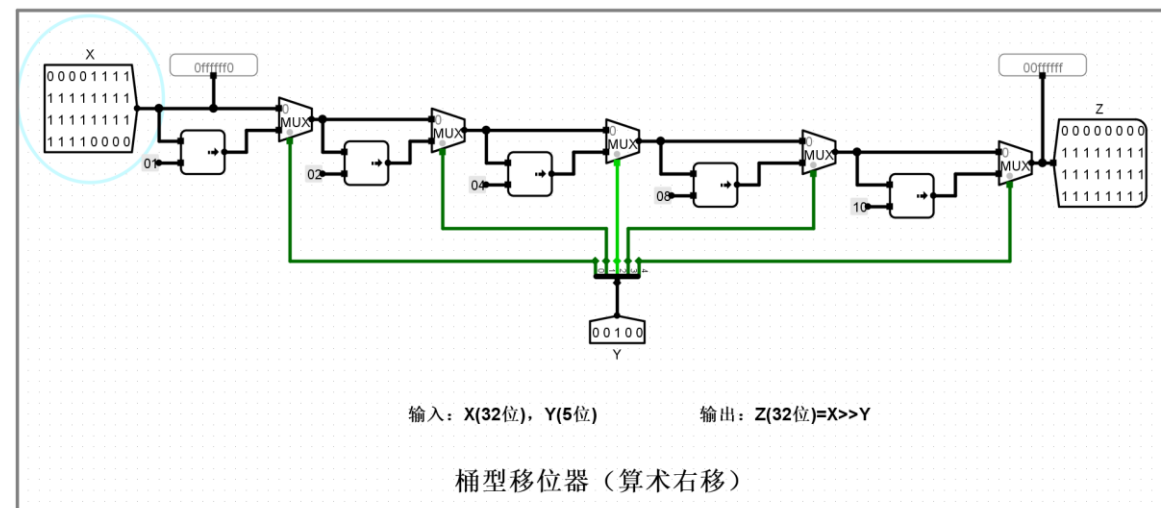
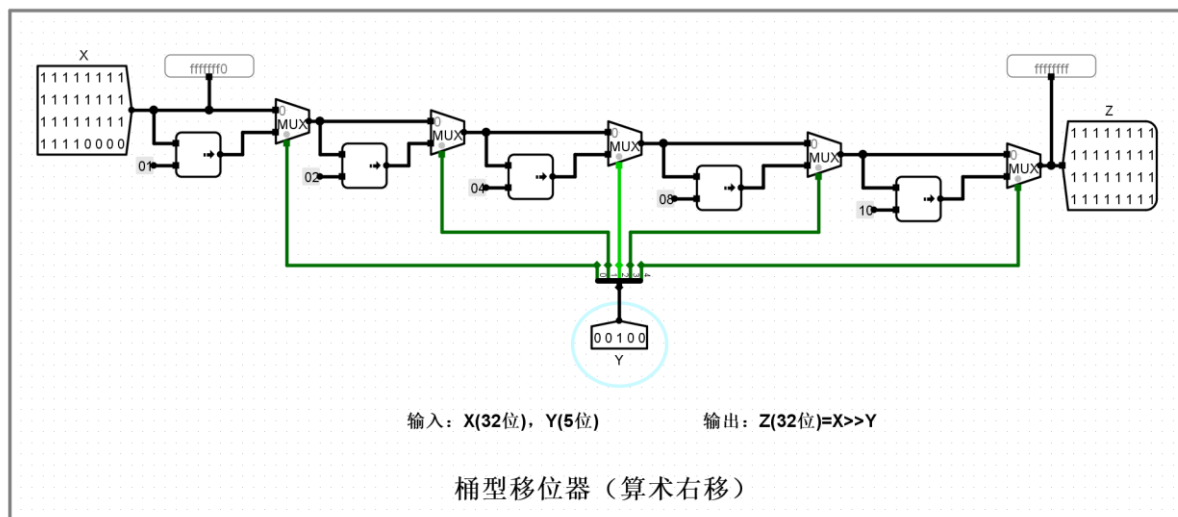


- 2、逻辑右移： $Z=X \gg Y$
 - 请验证： $X=\text{FFFFFFF0}$, $Y=00100$, $Z=\text{0FFFFFFF}$



• 3、算术右移： $Z=X \gg Y$

- 请验证： $X=\text{FFFFFFF0}$, $Y=00100$, $Z=\text{FFFFFFF}$
- 请验证： $X=\text{0FFFFFFF}$, $Y=00100$, $Z=\text{00FFFFFF}$



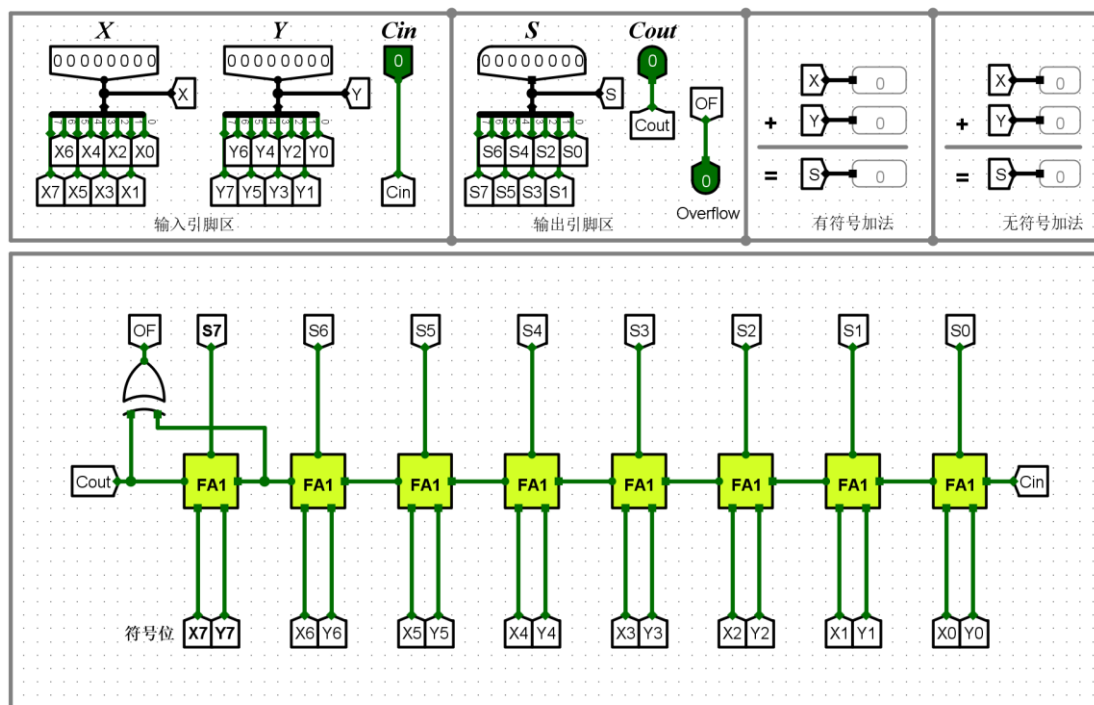
二、串行加法器

对应教材3.2.3小节（1、2、3）

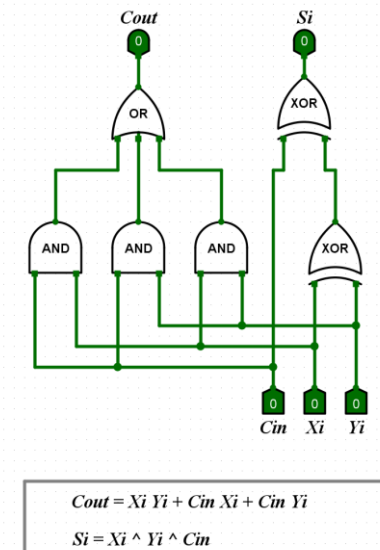
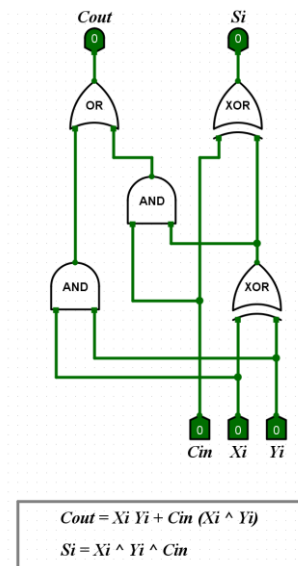
参考设计文件：串行加法器.circ（请从FTP下载）

二、串行加法器

- 1、8位串行加法器（教材图3.3）
 - 输入：X（8位）、Y（8位）、Cin（1位）
 - 输出：S（8位）、Cout（1位）、OF（溢出标志）

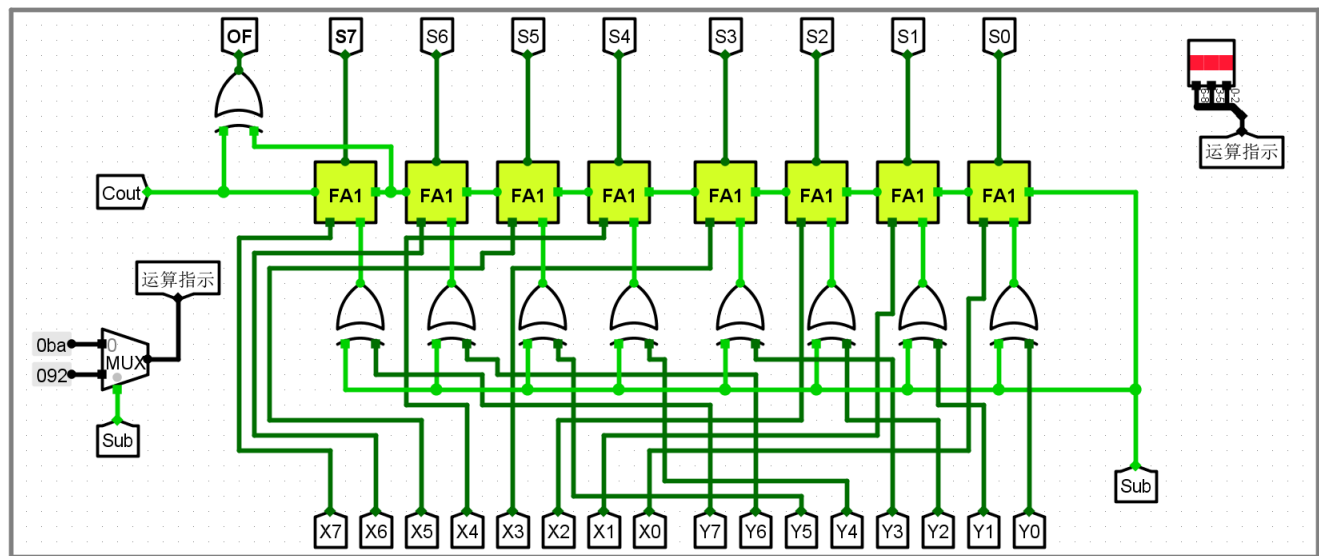
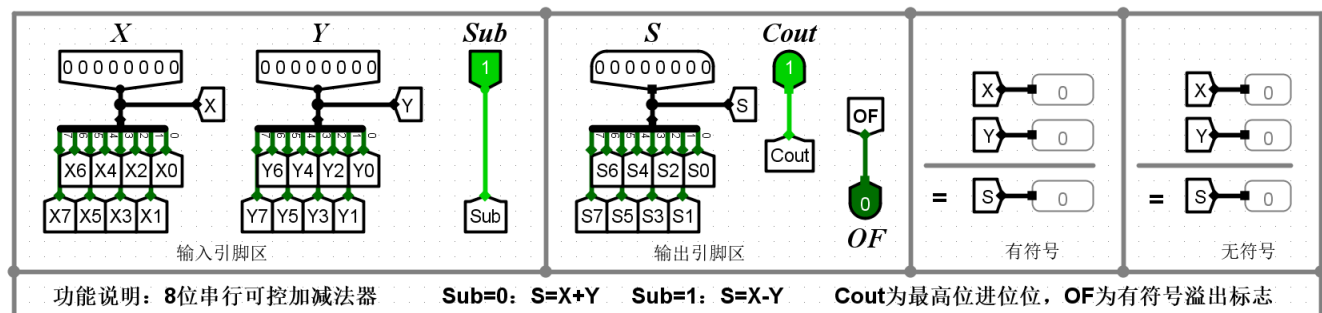


8位串行加法器 (1)

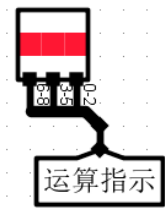
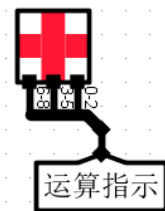


• 2、8位串行可控加减法器（教材图3.4）

- 输入：X（8位）、Y（8位）、Sub（1位；Sub=0，加法；Sub=1，减法）
- 输出：S（8位）、Cout（1位）、OF（溢出标志）、运算指示



8位串行可控加减法器（1）



- 请同学们通过设置不同的X、Y、Cin值，对上述8位串行加法器、8位串行可控加减法器（Sub=0）进行验证，并对结果进行分析（以下4种情况）：
 - 正数+正数=正数（不溢出）
 - 正数+正数=负数（溢出）
 - 负数+负数=负数（不溢出）
 - 负数+负数=正数（溢出）
- 请同学们通过设置不同的X、Y值，对上述8位串行可控加减法器（Sub=1）进行验证，并对结果进行分析（以下4种情况）：
 - 正数-负数=正数（不溢出）
 - 正数-负数=负数（溢出）
 - 负数-正数=负数（不溢出）
 - 负数-正数=正数（溢出）

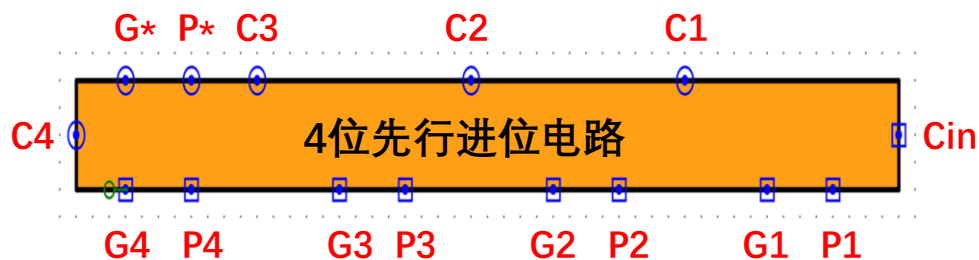
三、先行进位加法器

对应教材3.2.3小节（4）

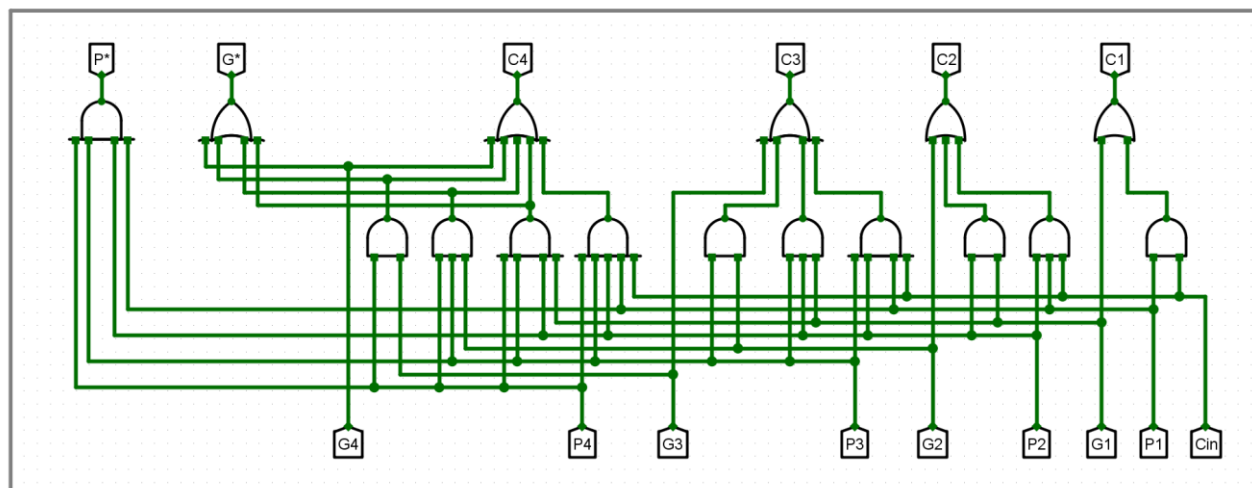
参考设计文件：先行进位加法器.circ（请从FTP下载）

三、先行进位加法器

- 1、4位先行进位电路（教材图3.5）
 - 输入：P1、G1、P2、G2、P3、G3、P4、G4、Cin
 - 输出：C1、C2、C3、C4、P*、G*



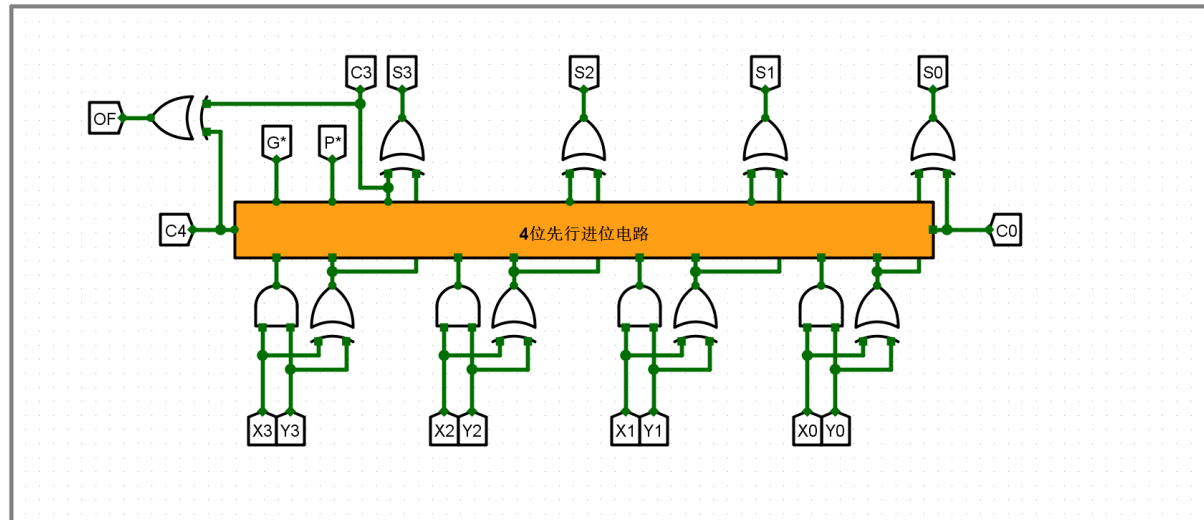
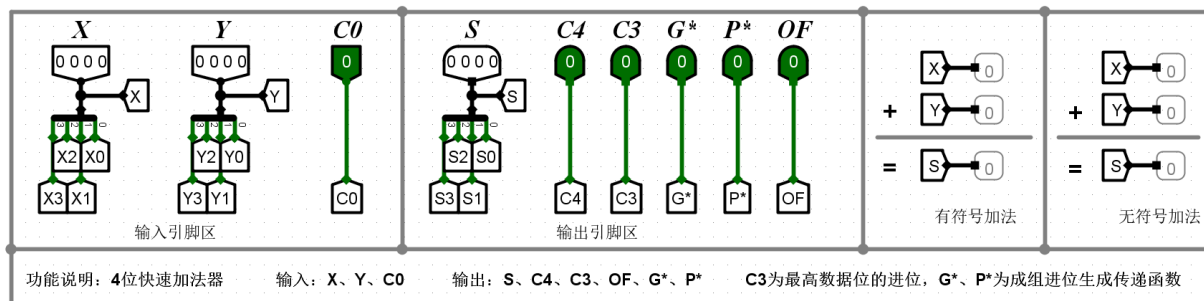
封装后的4位先行进位电路



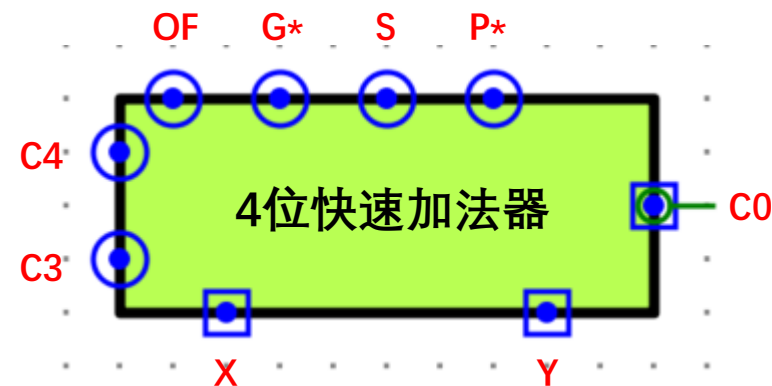
4位先行进位电路

• 2、4位快速加法器（教材图3.6）

- 输入：X（4位）、Y（4位）、C0
- 输出：S（4位）、C4、C3（最高数据位的进位，相当于公式3-5中的 C_d ）、OF、P*、G*



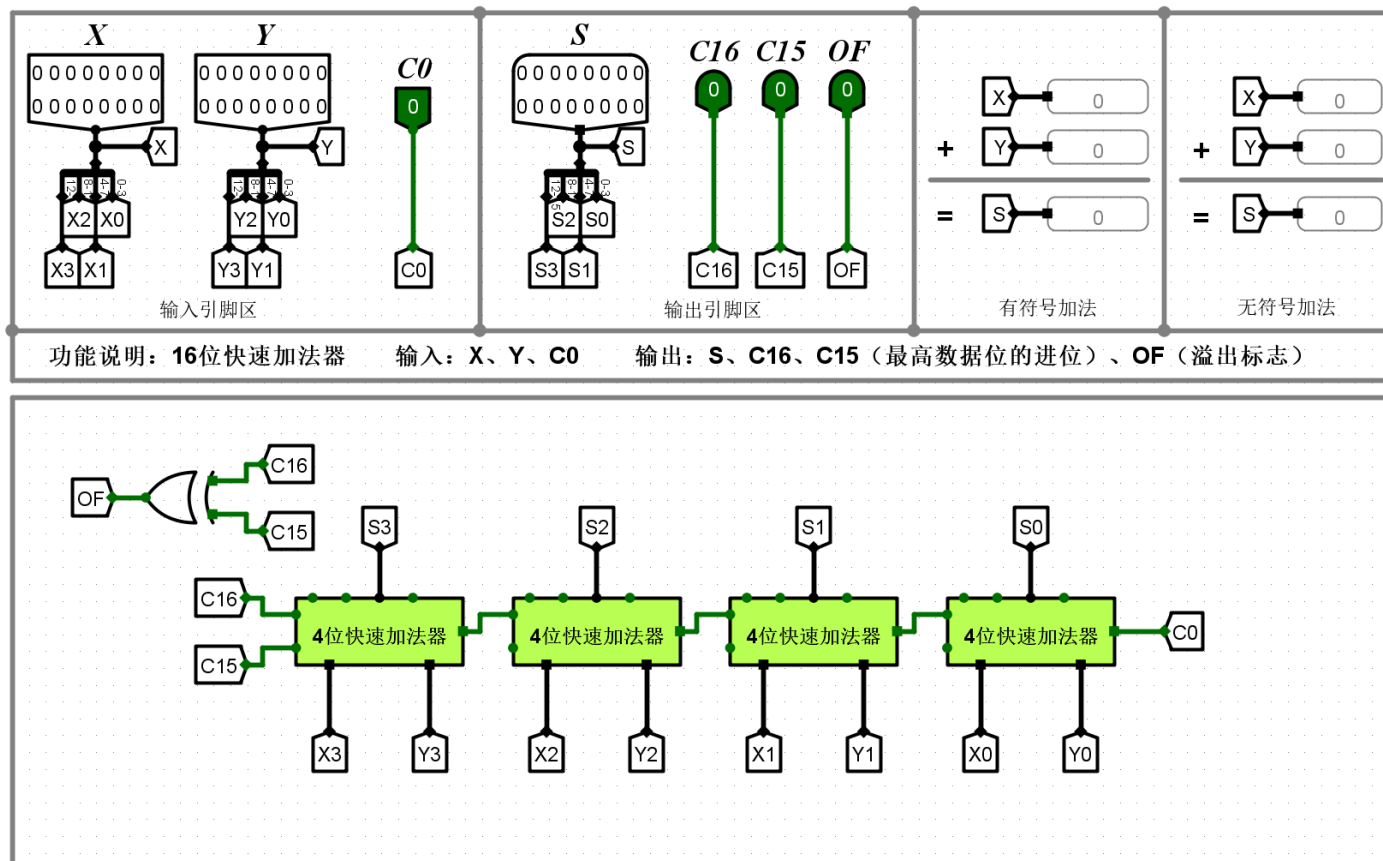
4位快速加法器



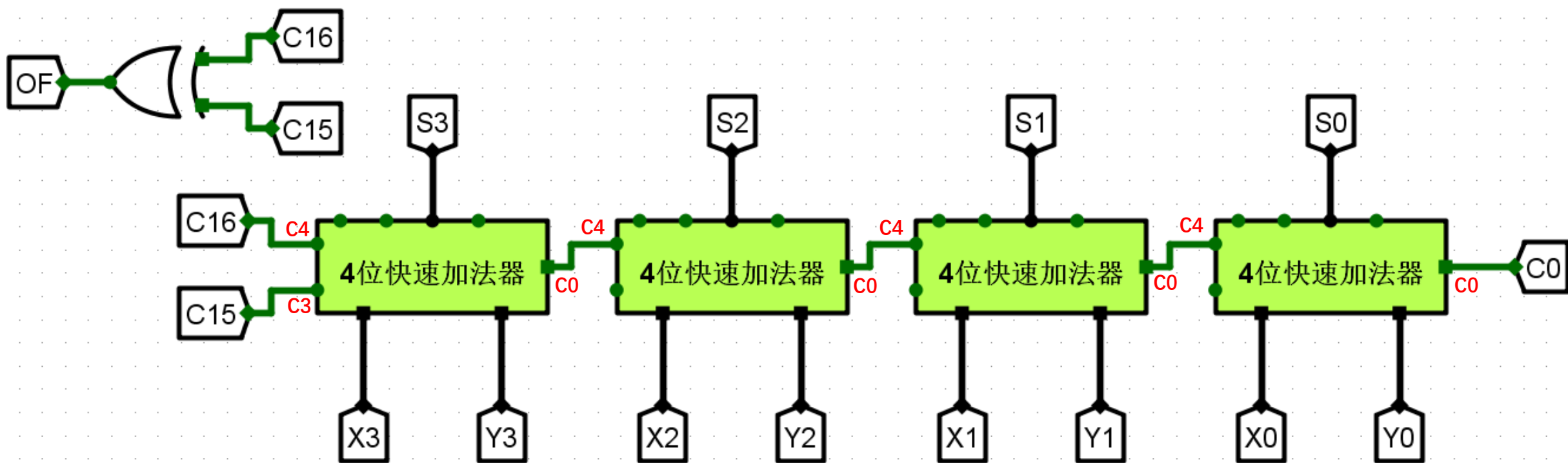
封装后的4位快速加法器

• 3、16位快速加法器（组内并行、组间串行，教材图3.7）

- 输入：X（16位）、Y（16位）、C0
- 输出：S（16位）、C16、C15（最高数据位的进位，相当于公式3-5中的 C_d ）、OF（溢出标志）

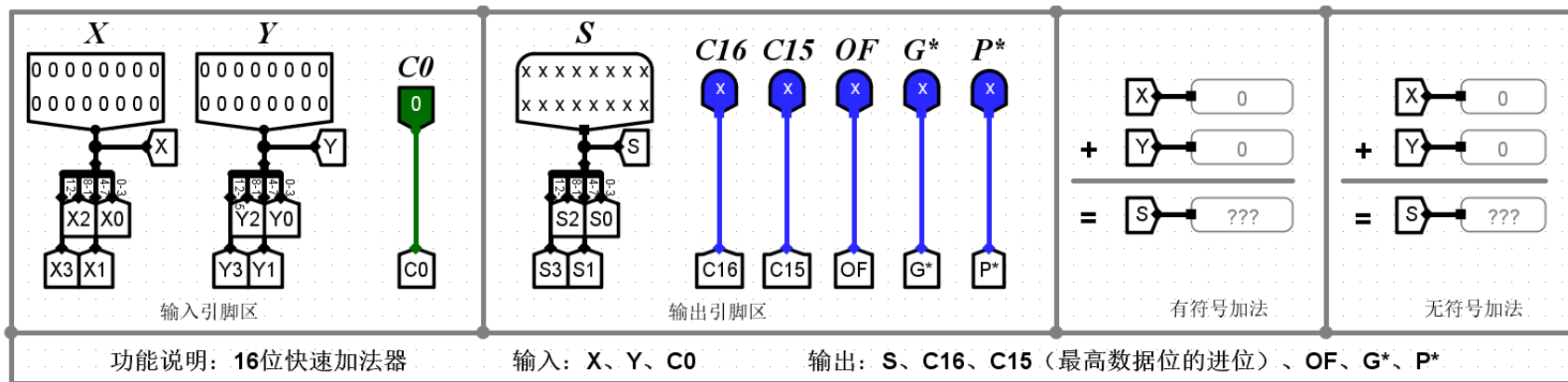


16位快速加法器（组内并行、组间串行）



16位快速加法器（组内并行、组间串行）

- 4、16位快速加法器（组内并行、组间并行，教材图3.8）（设计实验）
 - 输入：X（16位）、Y（16位）、C0
 - 输出：S（16位）、C16、C15（最高数据位的进位，相当于公式3-5中的 C_d ）、OF（溢出标志）、G*、P*
 - 请同学们设计该电路！

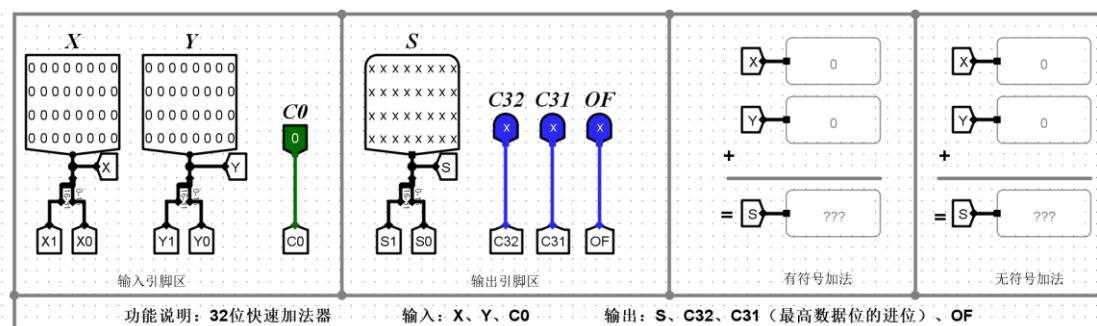


请同学们在此处设计电路！

16位快速加法器（组内并行、组间并行）

• 5、32位快速加法器（组内并行、组间串行）（设计实验）

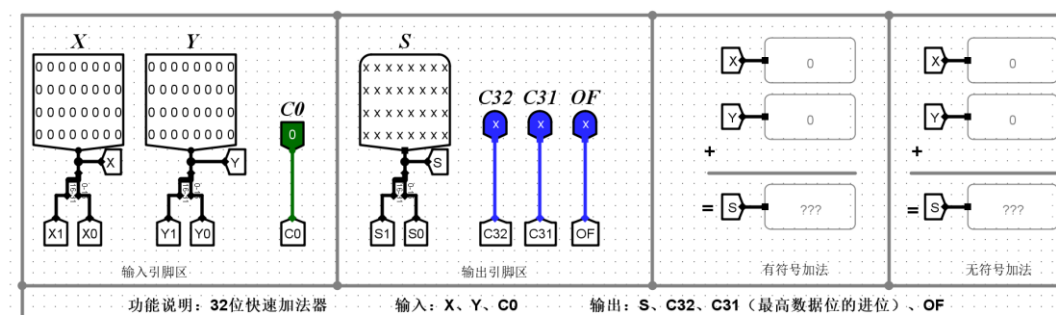
- 输入：X（32位）、Y（32位）、C0
- 输出：S（32位）、C32、C31（最高数据位的进位，相当于公式3-5中的 C_d ）、OF（溢出标志）
- 请同学们设计该电路！（16位加法器有两种电路）



请同学们在此处设计电路！

（16位加法器采用：组内并行、组间串行）

32位快速加法器（组内并行、组间串行）



请同学们在此处设计电路！

（16位加法器采用：组内并行、组间并行）

32位快速加法器（组内并行、组间串行）

- 请同学们通过设置不同的X、Y、C0值，对上述4位快速加法器、16位快速加法器（两种方式）、32位快速加法器（两种电路）进行验证，并对结果进行分析（以下4种情况）：
 - 正数+正数=正数（不溢出）
 - 正数+正数=负数（溢出）
 - 负数+负数=负数（不溢出）
 - 负数+负数=正数（溢出）

四、算术逻辑单元 (ALU)

对应教材3.6.1小节

参考设计文件：算术逻辑单元.circ（请从FTP下载）

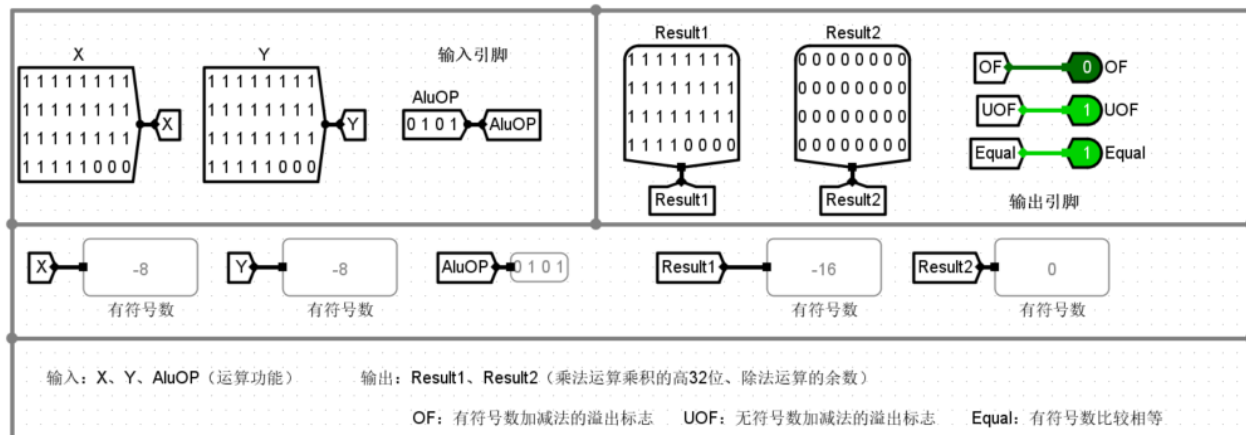
四、算术逻辑单元 (ALU)

- 32位ALU（算术逻辑单元）

- 输入：X（32位）、Y（32位）、AluOP（4位，决定ALU的运算功能）
- 输出：Result1（32位）、Result2（32位）、OF（有符号数加减法的溢出标志）、UOF（无符号数加减法的溢出标志）、Equal（有符号数比较相等标志）

ALU的运算功能（13种运算功能）

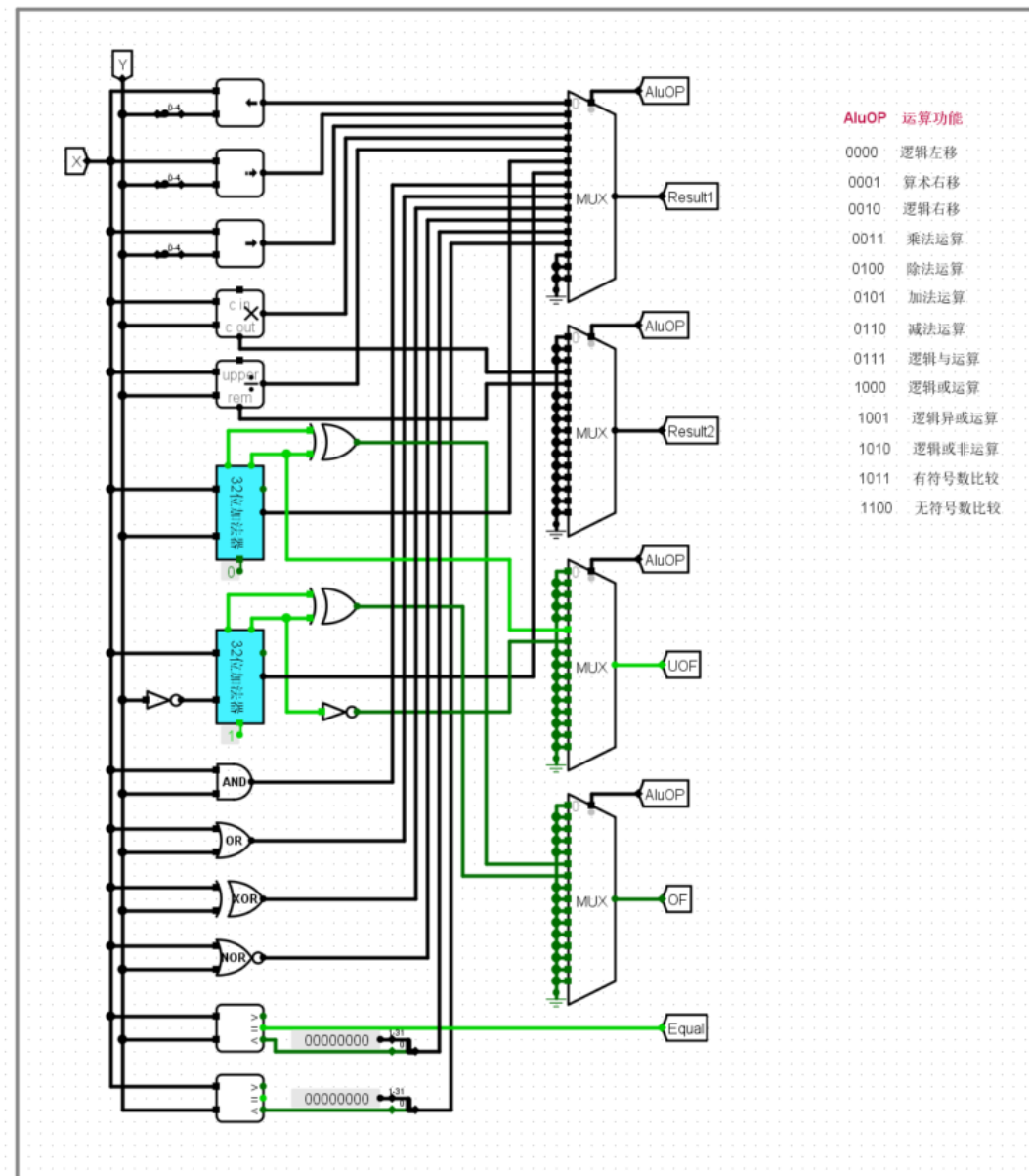
AluOP	十进制	运算功能	功能说明
0000	0	逻辑左移	Result1 = X<<Y（Y取低5位）；Result2=0
0001	1	算术右移	Result1 = X>>Y（Y取低5位）；Result2=0
0010	2	逻辑右移	Result1 = X>>Y（Y取低5位）；Result2=0
0011	3	乘法	Result1 = (X*Y) _[31:0] ；Result2 = (X*Y) _[63:32]
0100	4	除法	Result1 = X/Y（商）；Result2 = X%Y（余数）
0101	5	加法	Result1 = X+Y（置OF/UOF标志）；Result2=0
0110	6	减法	Result1 = X-Y（置OF/UOF标志）；Result2=0
0111	7	逻辑与	Result1 = X&Y；Result2=0
1000	8	逻辑或	Result1 = X Y；Result2=0
1001	9	逻辑异或	Result1 = X⊕Y；Result2=0
1010	10	逻辑或非	Result1 = ~(X Y)；Result2=0
1011	11	有符号数比较	Result1 = (X<Y) ? 1 : 0（X、Y为有符号数；如果相等，则Equal=1）；Result2=0
1100	12	无符号数比较	Result1 = (X<Y) ? 1 : 0（X、Y为无符号数）；Result2=0



减法运算: $\text{Result1} = X - Y = X + /Y + 1$

加法运算: $\text{OF} = \text{C32} \oplus \text{C31}$, $\text{UOF} = \text{C32}$

减法运算: $\text{OF} = \text{C32} \oplus \text{C31}$, $\text{UOF} = \neg \text{C32}$



32位ALU (算术逻辑单元)

- 请同学们在上述32位ALU中，通过改变不同的X、Y、AluOP值，完成下列运算功能：

- ① 逻辑左移
- ② 算术右移
- ③ 逻辑右移
- ④ 乘法运算
- ⑤ 除法运算
- ⑥ 加法运算
- ⑦ 减法运算
- ⑧ 逻辑与运算
- ⑨ 逻辑或运算
- ⑩ 逻辑异或运算
- ⑪ 逻辑或非运算
- ⑫ 有符号数比较
- ⑬ 无符号数比较

- 并对结果进行分析！（重点分析加减法运算中的OF、UOF，以及有符号数比较和无符号数比较）

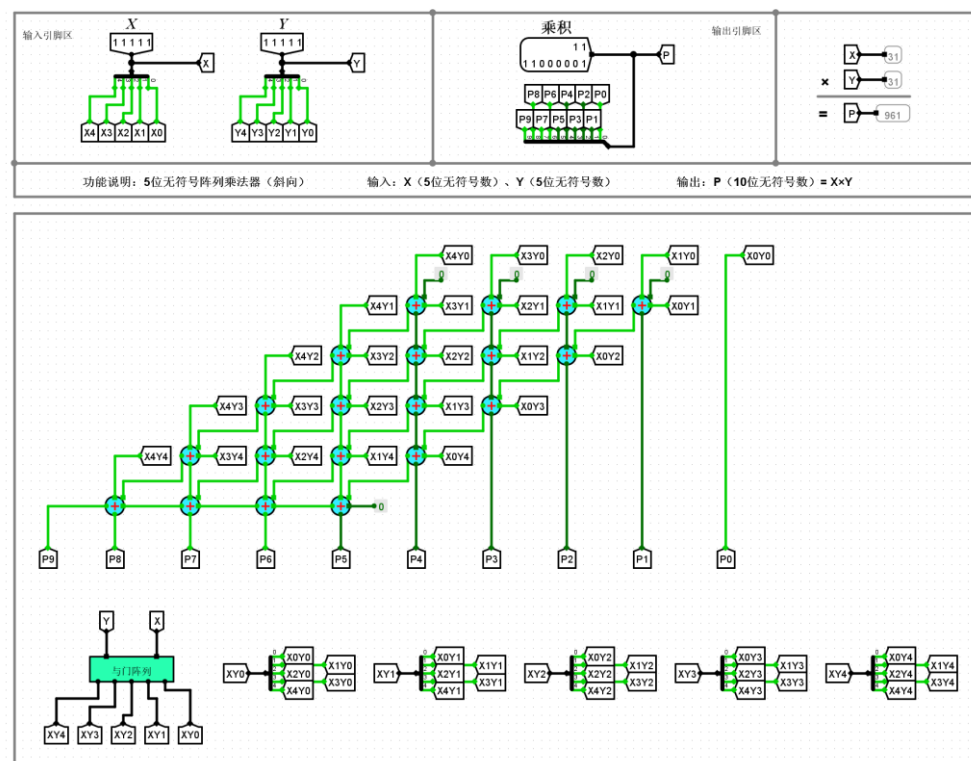
五、阵列乘法器

对应教材3.3.3小节和3.3.4小节

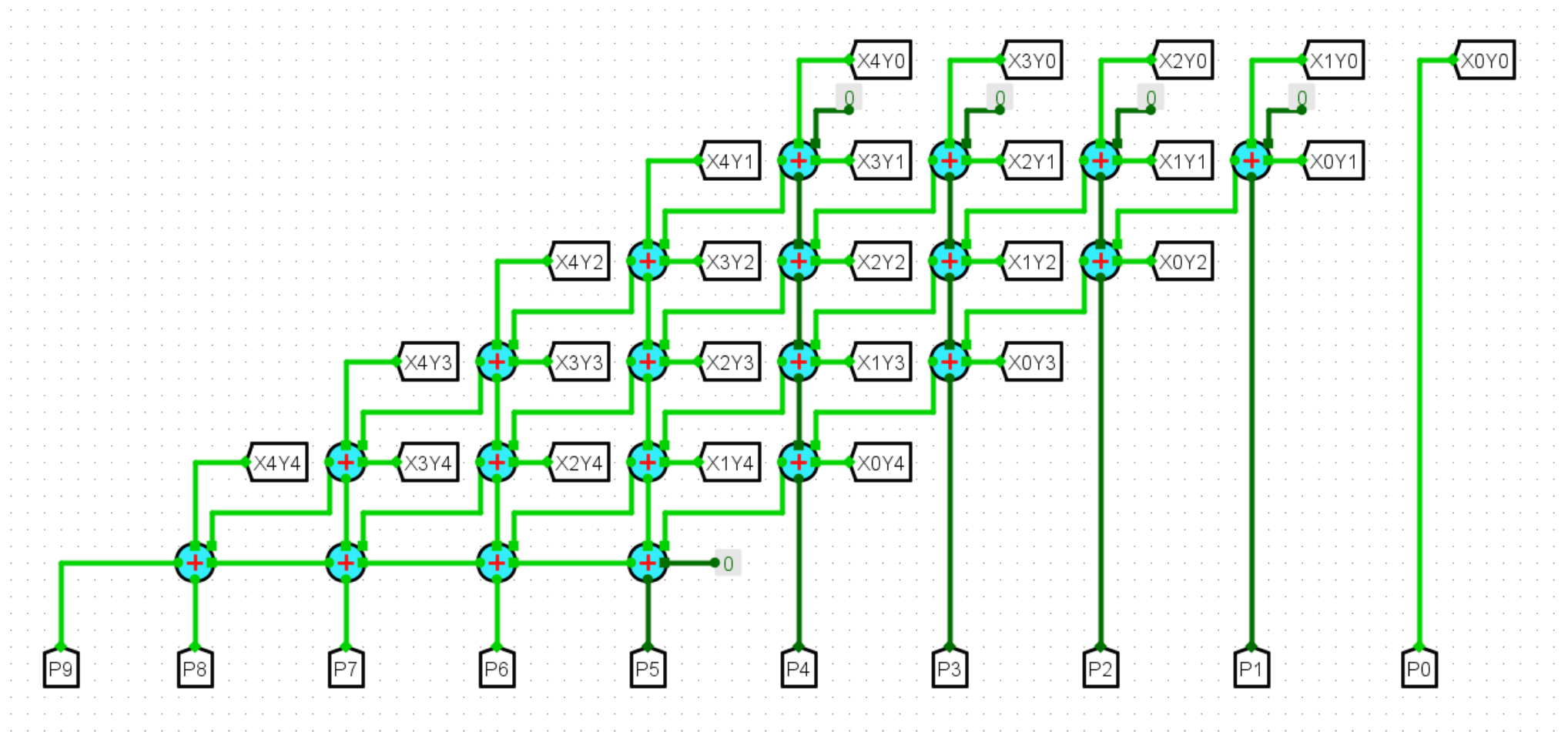
参考设计文件：阵列乘法器.circ（请从FTP下载）

五、阵列乘法器

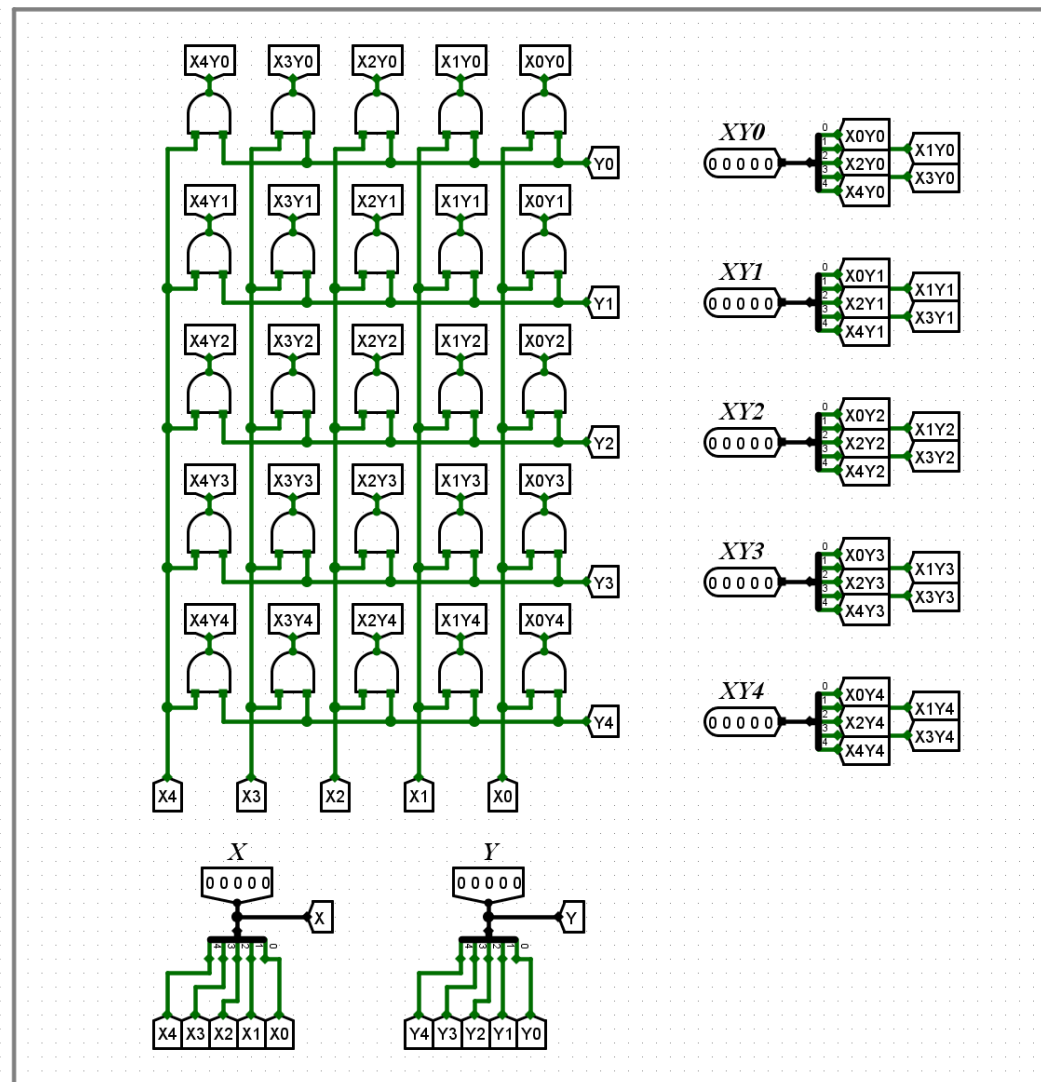
- 1、5位无符号阵列乘法器（斜向，对应教材的图3.13）
 - 输入：X（5位无符号数）、Y（5位无符号数）
 - 输出：P（10位无符号数）



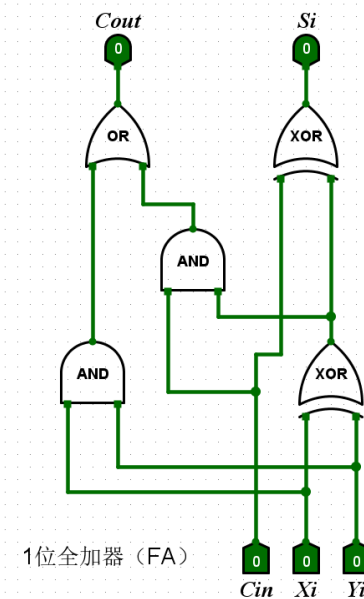
5位无符号阵列乘法器（斜向）



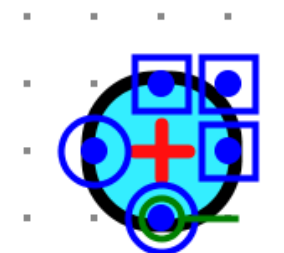
5位无符号阵列乘法器（斜向）



与门阵列（5位）

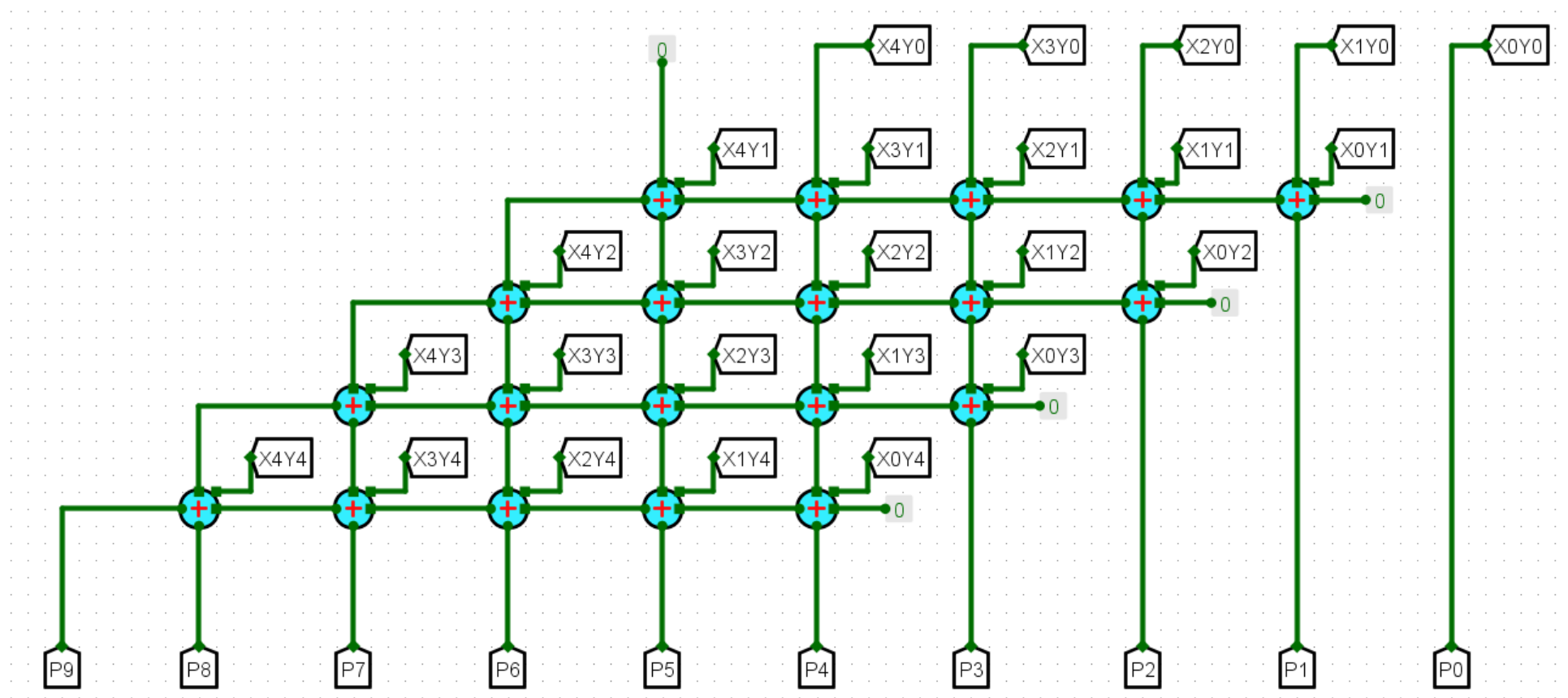


1位全加器（FA）



封装后的1位全加器

- 2、5位无符号阵列乘法器（横向）
 - 输入：X（5位无符号数）、Y（5位无符号数）
 - 输出：P（10位无符号数）



5位无符号阵列乘法器（横向）

- 请同学们通过改变X、Y的值，对上述5位无符号阵列乘法器（斜向、横向）进行验证：
 - $X \neq 0, Y \neq 0, P = ?$
 - $X = 0, Y \neq 0, P \text{ 是不是 } = 0?$
 - $X \neq 0, Y = 0, P \text{ 是不是 } = 0?$
 - $X = 0, Y = 0, P \text{ 是不是 } = 0?$
 - $X = 31, Y = 31, P \text{ 是不是 } = 961?$

• 3、6位原码阵列乘法器（设计实验）

- 输入（6位）：X= $X_5, X_4, X_3, X_2, X_1, X_0$ ，Y= $Y_5, Y_4, Y_3, Y_2, Y_1, Y_0$ ；X=-31~+31，Y=-31~+31
- 输出（11位）：P= $P_{10}, P_9, P_8, P_7, P_6, P_5, P_4, P_3, P_2, P_1, P_0$ ；P=-961~+961
- 其中， X_5 、 Y_5 、 P_{10} 为符号位

• 请同学们利用前面的5位无符号阵列乘法器（可以是斜向，也可以是横向），实现6位原码阵列乘法器

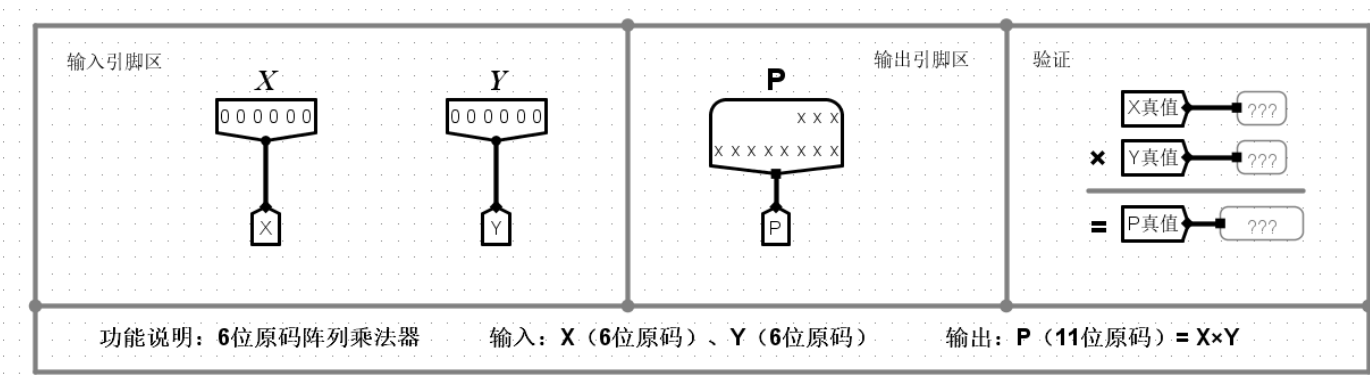
• 思路：

- 数值位（5位）采用5位无符号阵列乘法器，符号位单独处理
- 令： $|X|$ （5位）= X_4, X_3, X_2, X_1, X_0 ； $|Y|$ （5位）= Y_4, Y_3, Y_2, Y_1, Y_0
- 则： $X = X_5, |X|$ ； $Y = Y_5, |Y|$
- 将 $|X|$ 和 $|Y|$ 通过5位无符号阵列乘法器进行运算，得到10位无符号乘积 $|P| = P_9, P_8, P_7, P_6, P_5, P_4, P_3, P_2, P_1, P_0$
- 则： $P = P_{10}, |P|$
- 其中： $P_{10} = X_5 \oplus Y_5$

• 验证：请同学们通过改变X、Y的值，对实现的6位原码阵列乘法器进行验证：

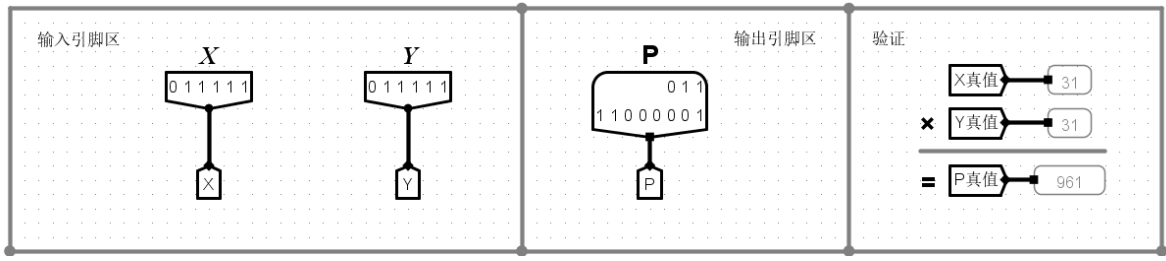
- X=正数，Y=正数，P=正数（例如：X=31、Y=31、P=961）
- X=正数，Y=负数，P=负数（例如：X=31、Y=-31、P=-961）
- X=负数，Y=正数，P=负数（例如：X=-31、Y=31、P=-961）
- X=负数，Y=负数，P=正数（例如：X=-31、Y=-31、P=961）

• 注意：需要将X、Y、P的原码转换为真值进行显示

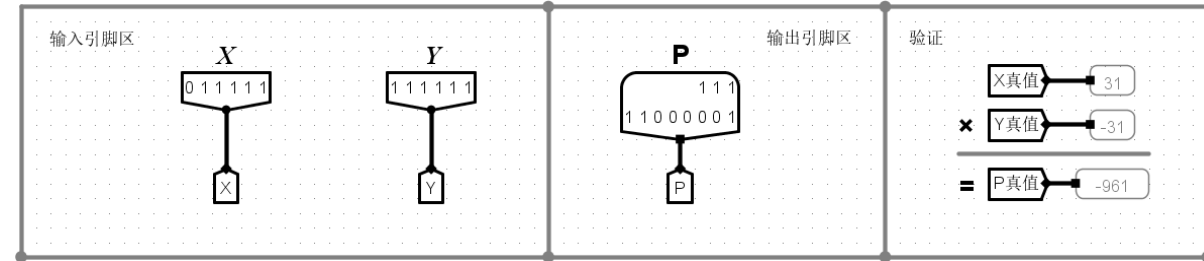


除了要得到原码的乘积P，还要将运算结果用真值显示出来，即要将原码（X、Y、P）转换为真值（X真值、Y真值、P真值）

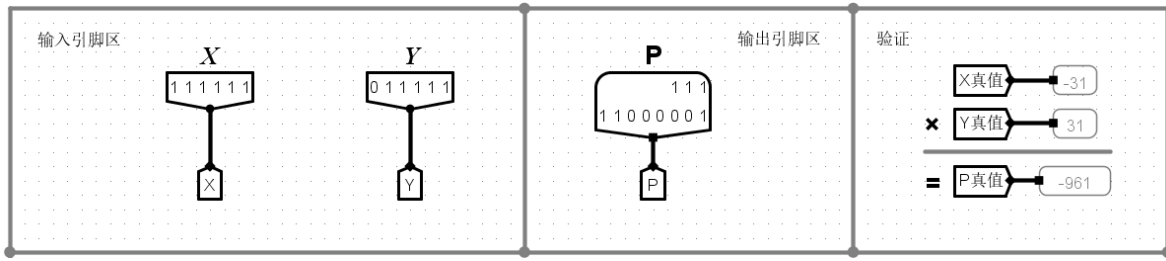
正数 × 正数 = 正数



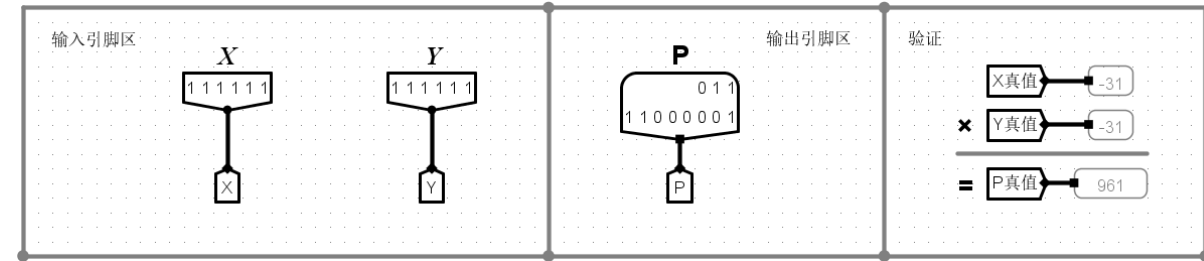
正数 × 负数 = 负数



负数 × 正数 = 负数



负数 × 负数 = 正数



• 4、6位补码阵列乘法器（设计实验）

- 输入（6位）：X=X5,X4X3X2X1X0，Y=Y5,Y4Y3Y2Y1Y0；X=-32~+31；Y=-32~+31；
- 输出（12位）：P=P11,P10P9P8P7P6P5P4P3P2P1P0；P=-992~+1024；考虑到P的最大值为+1024，超出11位二进制数的表示范围，因此P用12位二进制数表示

• 请同学们利用前面的5位无符号阵列乘法器（可以是斜向，也可以是横向），实现6位补码阵列乘法器

• 思路：

- 请参考教材中的图3.14
- 如果X、Y为负数，先采用求补器将X、Y转换为5位无符号数，然后再采用5位无符号数阵列乘法器计算乘积
- 如果是正数与负数相乘，则需要通过求补器将乘积转换为补码

• 下列特殊情况，会产生错误的结果，需要单独处理（设计附加电路）：

- ①X=0，Y=负数；P=-1024；错误
- ②X=负数，Y=0；P=-1024；错误
- ③X=-32，Y=0或正数；P=-1024；错误
- ④X=0或正数，Y=-32；P=-1024；错误
- ⑤X=-32，Y=负数；P=0；错误
- ⑥X=负数，Y=-32；P=0；错误

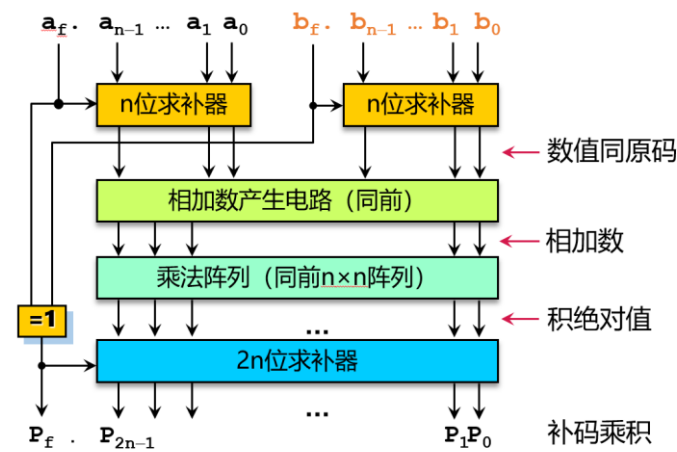
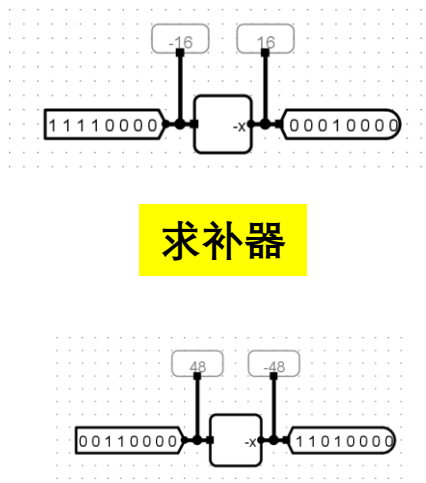


图3.14 补码阵列乘法器



- 请同学们通过改变X、Y的值，对上述**6位补码阵列乘法器**进行验证：

① $X=0$ ， Y =负数； $P=0$

② X =负数， $Y=0$ ； $P=0$

③ $X=-32$ ， $Y=0$ 或正数； P =正确的数

④ $X=0$ 或正数， $Y=-32$ ； P =正确的数

⑤ $X=-32$ ， Y =负数； P =正确的数

⑥ X =负数， $Y=-32$ ； P =正确的数

⑦ $X=-32$ ， $Y=-32$ ， $P=1024$

⑧ $X=31$ ， $Y=31$ ， $P=961$

⑨ $X=-32$ ， $Y=31$ ， $P=-992$

⑩ $X=31$ ， $Y=-32$ ， $P=-992$

⑪ X =正数， Y =正数； P =正确的数

⑫ X =正数， Y =负数； P =正确的数

⑬ X =负数， Y =正数； P =正确的数

⑭ X =负数， Y =负数； P =正确的数

六、原码和补码一位乘法器

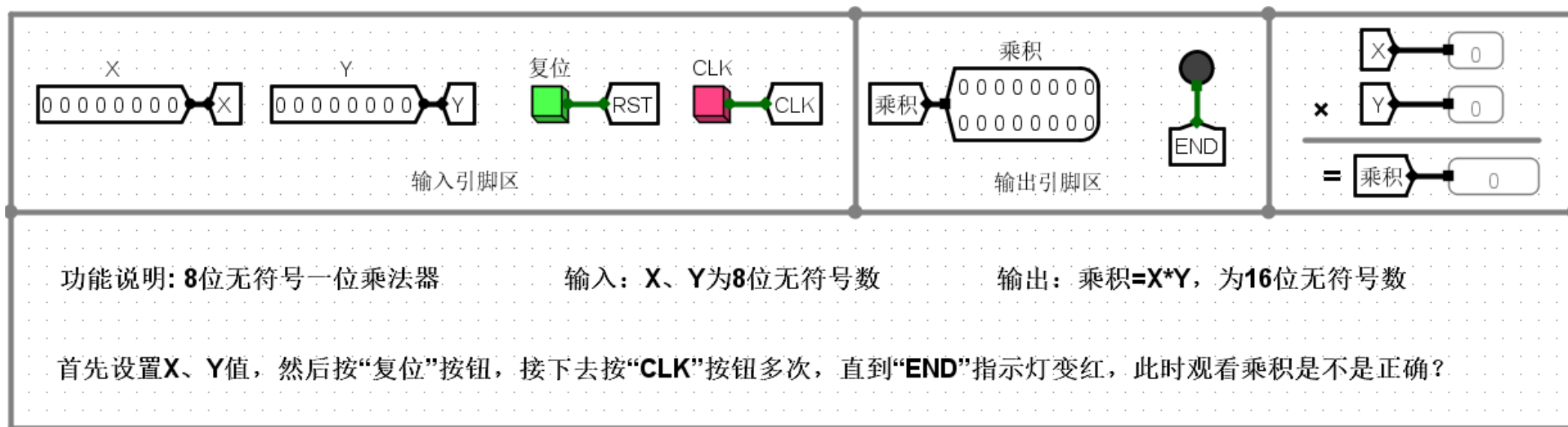
对应教材3.3.1小节和3.3.2小节

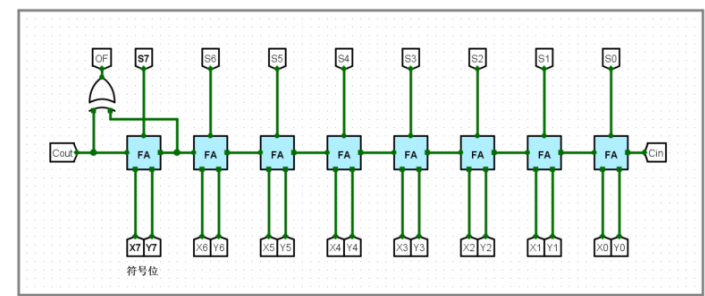
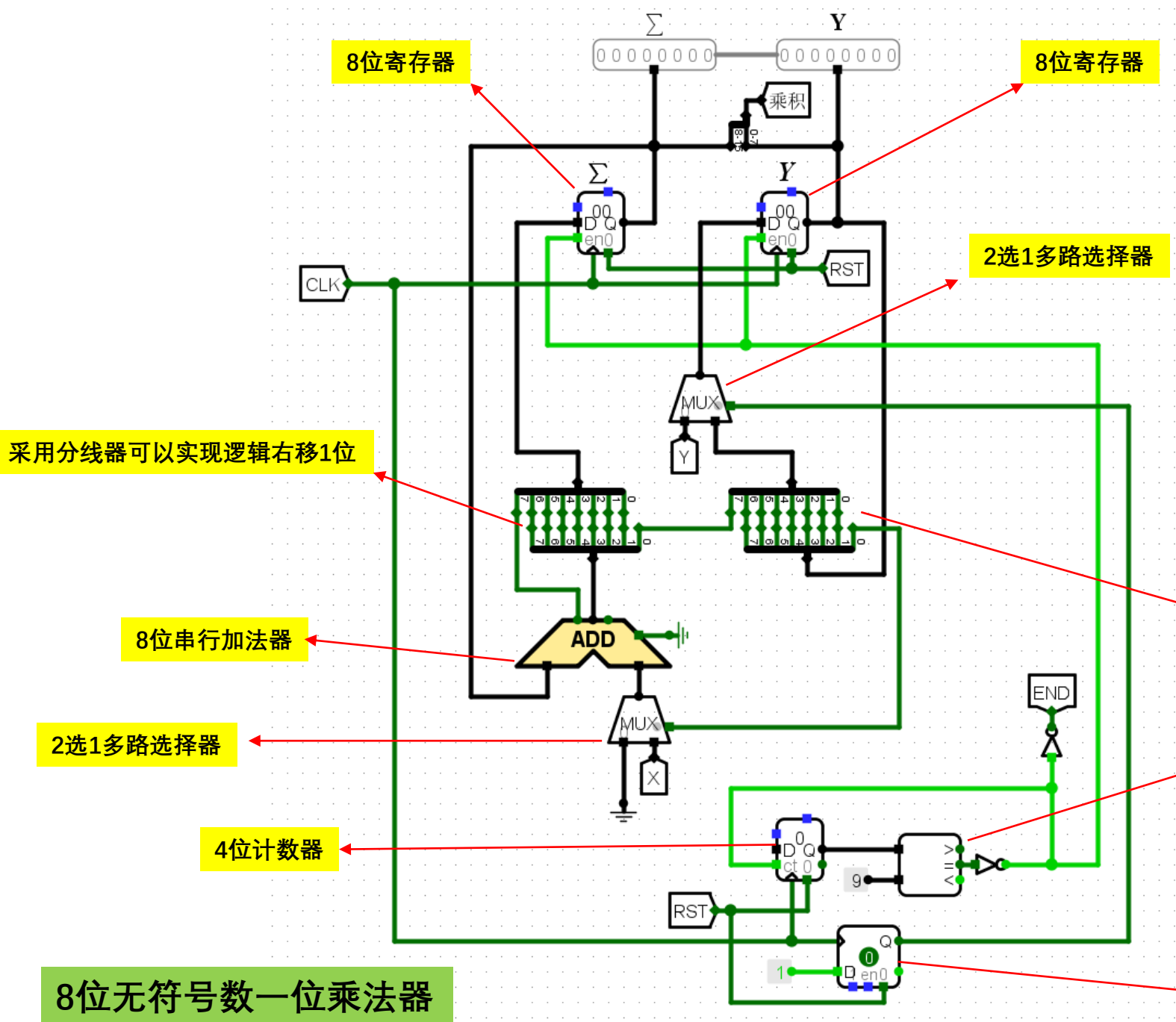
参考设计文件：原码和补码一位乘法器.circ（请从FTP下载）

六、原码和补码一位乘法器

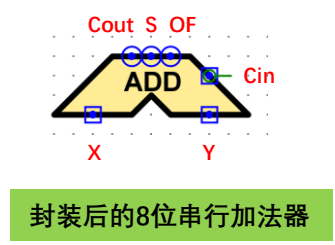
• 1、8位无符号数一位乘法器

- 输入：X（8位，无符号数）、Y（8位，无符号数）、复位（RST）、时钟（CLK）；X=0~255，Y=0~255
- 输出：乘积（16位，无符号数）、运算结束指示灯（END），乘积=0~65025





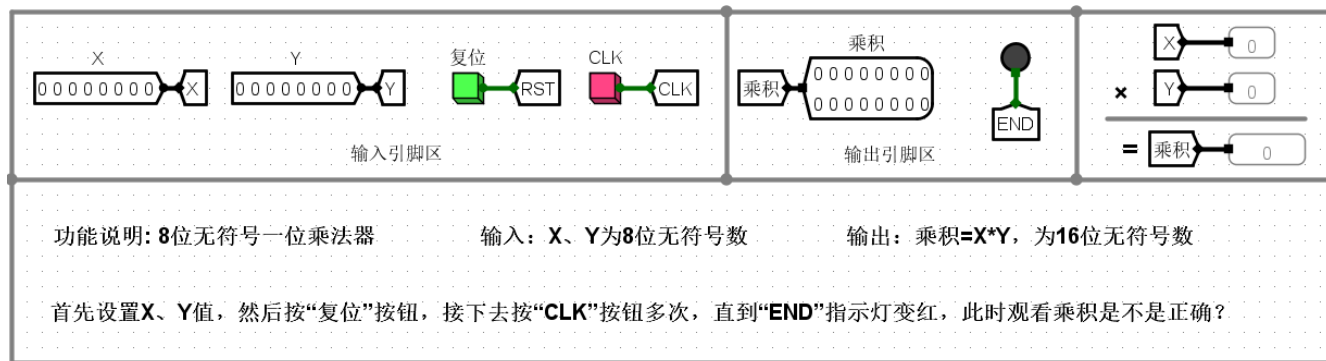
8位串行加法器



采用分线器可以实现逻辑右移1位

- 8位无符号数一位乘法器的运行:

- (1) 设置X和Y的值
- (2) 鼠标点击电路图中的“复位”
- (3) 鼠标点击电路图中的“CLK”若干次, 直到运算结束指示灯 (END) 变红
- (4) 观看运算结果 (乘积) 是否正确



- 请同学们设置不同的X、Y值 (无符号数)

- X=0, Y=任意, 乘积是否=0
- X=任意, Y=0, 乘积是否=0
- X=255, Y=255, 乘积是否=65025
- X=其他, Y=其他, 乘积是否正确?

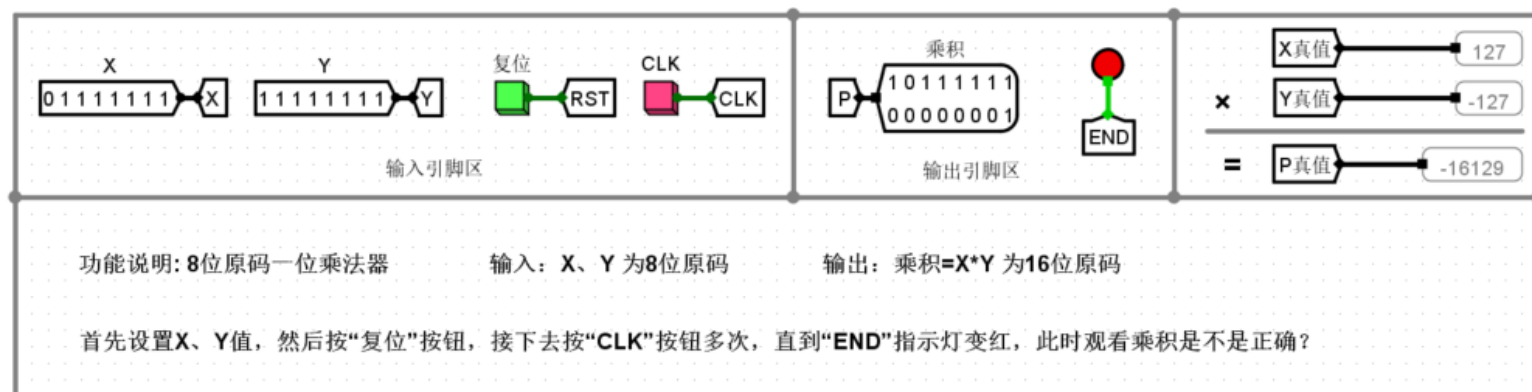
• 2、8位原码一位乘法器（设计实验）

- 输入：X（8位，原码）、Y（8位，原码）、复位（RST）、时钟（CLK）
- 输出：P（16位，原码）、运算结束指示灯（END）、X的真值、Y的真值、P的真值

• 请同学们根据前面的8位无符号数一位乘法器，设计8位原码一位乘法器

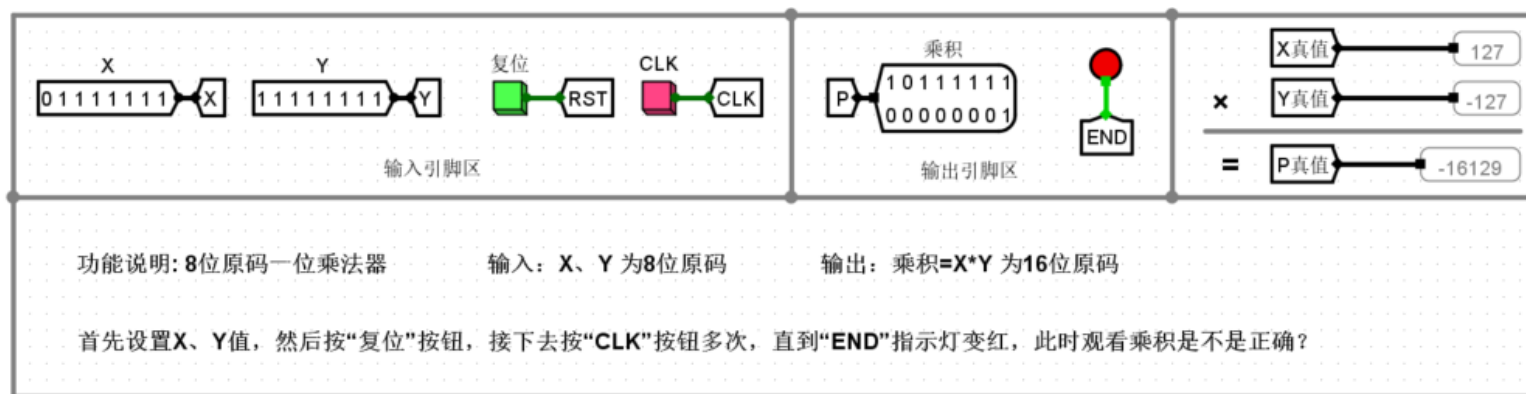
- **思路：**首先将8位原码（X和Y）转换为8位无符号数（绝对值， $|X|$ 和 $|Y|$ ），然后运用8位无符号数一位乘法器得到乘积（绝对值， $|P|$ ），再将乘积的绝对值转换为原码；乘积的符号位： $P_{15}=X_7 \oplus Y_7$

- **注意：**需要将X、Y、P的原码转换为**真值**进行显示



- 8位原码一位乘法器的运行：

- (1) 设置X和Y的值
- (2) 鼠标点击电路图中的“复位”
- (3) 鼠标点击电路图中的“CLK”若干次，直到运算结束指示灯（END）变红
- (4) 观看运算结果（P的真值）是否正确

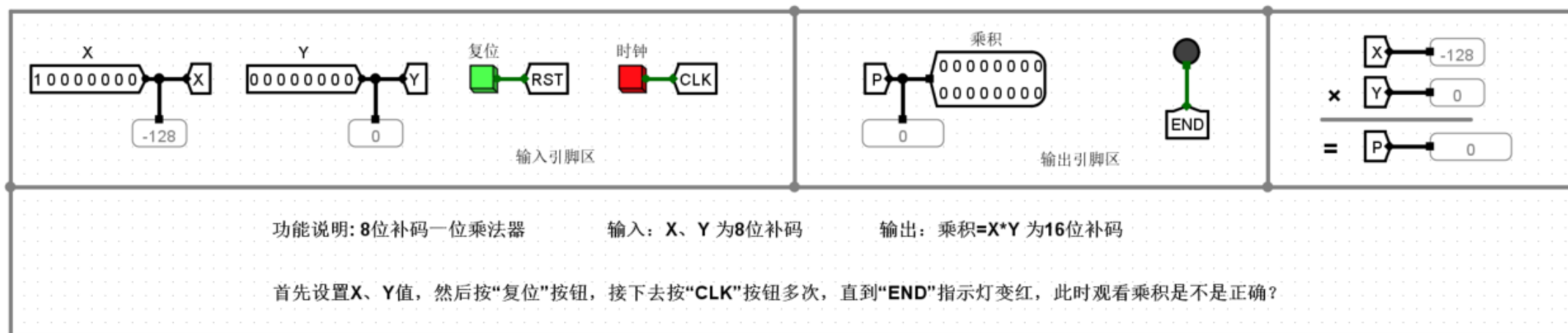


- 请同学们设置不同的X、Y值（原码）

- X=0, Y=任意, P是否=0
- X=任意, Y=0, P是否=0
- X=127, Y=127, P是否=16129
- X=-127, Y=127, P是否=-16129
- X=127, Y=-127, P是否=-16129
- X=-127, Y=-127, P是否=16129
- X=其他, Y=其他, P是否正确?

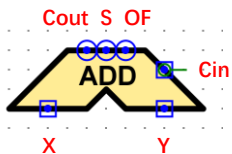
• 3、8位补码一位乘法器

- 输入：X（8位，补码）、Y（8位，补码）、复位（RST）、时钟（CLK）
- 输出：乘积（16位，补码）、运算结束指示灯（END）



8位补码一位乘法器电路

采用分线器可以实现算术右移1位

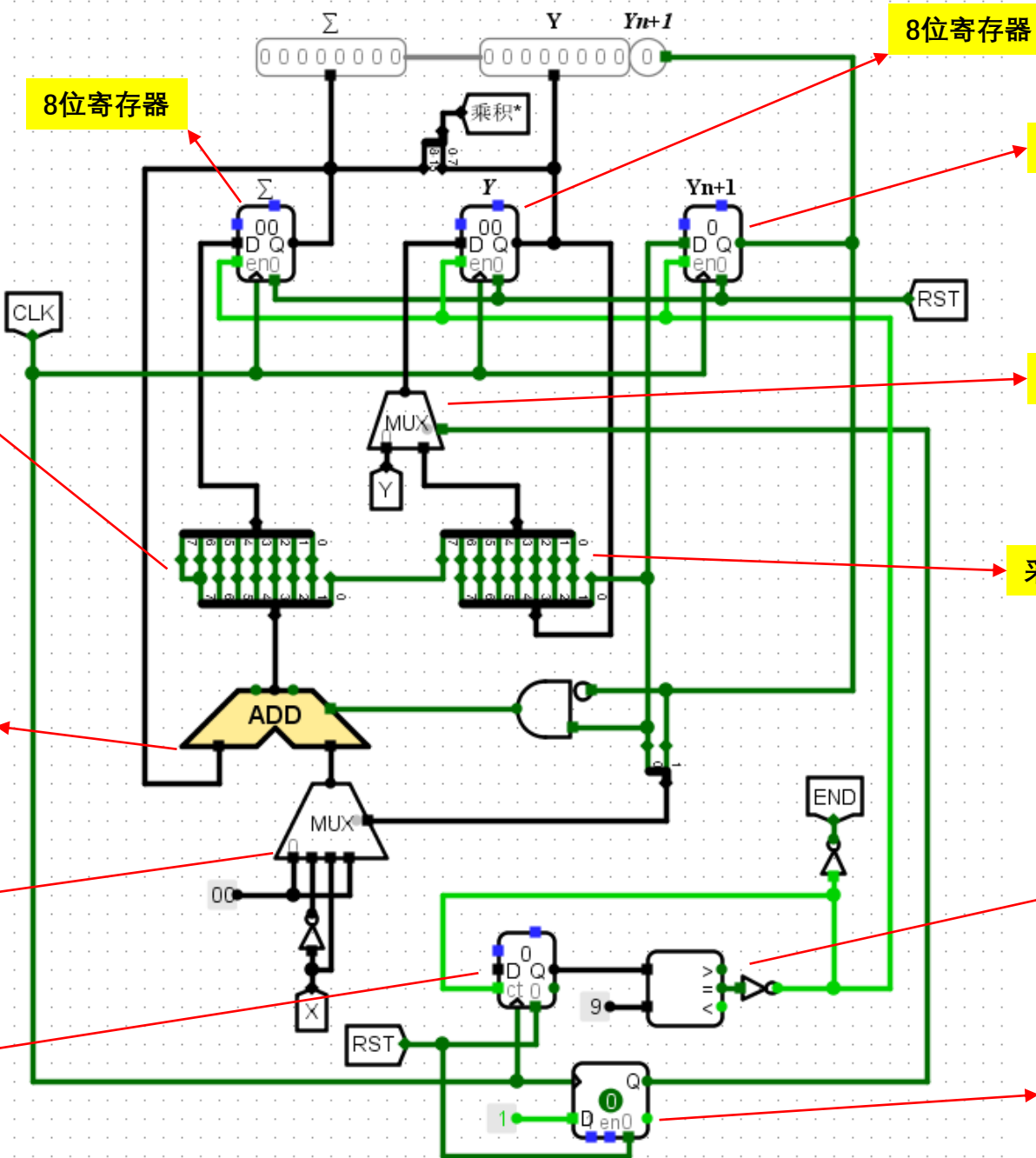


封装后的8位串行加法器

8位串行加法器

4选1多路选择器

4位计数器



8位寄存器

1位寄存器

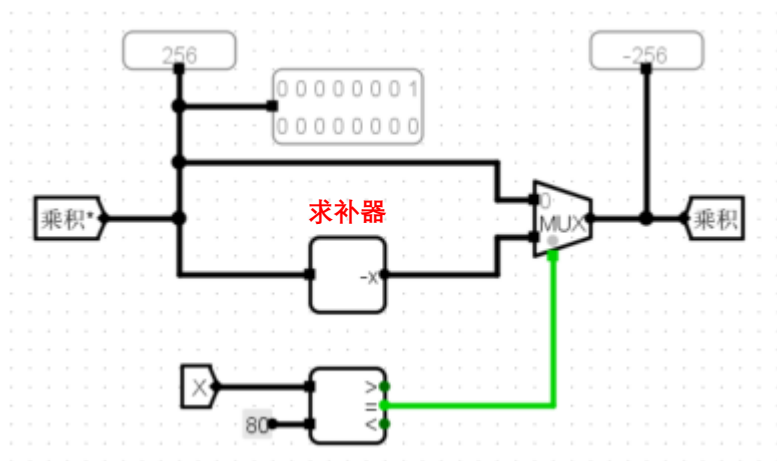
2选1多路选择器

采用分线器可以实现算术右移1位

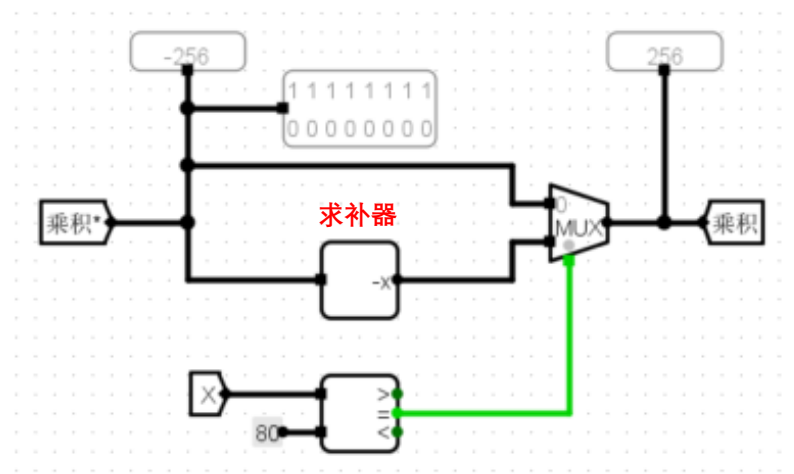
4位无符号数比较器

D触发器

- 采用这个8位补码一位乘法器计算时，会存在以下的问题：
 - 当 $X=1000\ 0000=-128$ 时，此时Y为正数（如 $Y=0000\ 0010=2$ ），得到的结果 $=00000001\ 00000000=256$ ，真是的结果 $= -256=11111111\ 00000000$
 - 当 $X=1000\ 0000=-128$ 时，此时Y为负数（如 $Y=1111\ 1110=-2$ ），得到的结果 $=11111111\ 00000000=-256$ ，真是的结果 $=256=00000001\ 00000000$
 - 因此，需要通过附加的电路（**简化版**），对运算结果进行处理，具体如下：



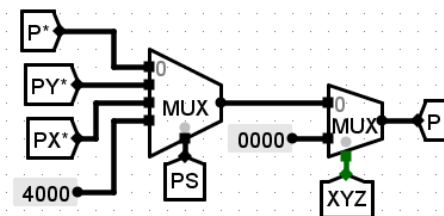
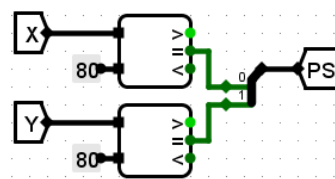
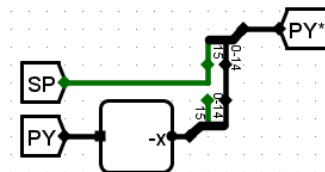
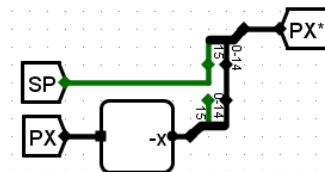
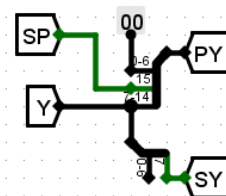
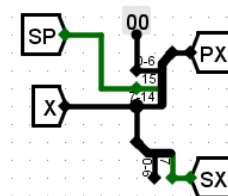
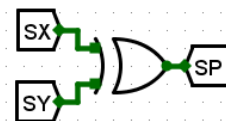
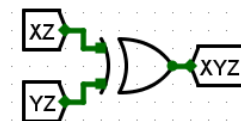
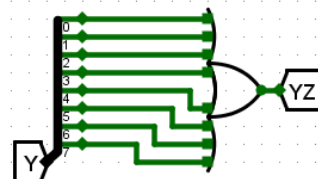
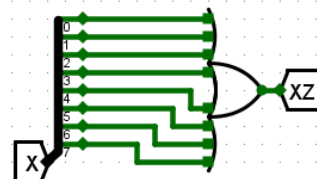
80H=1000 0000=-128



80H=1000 0000=-128

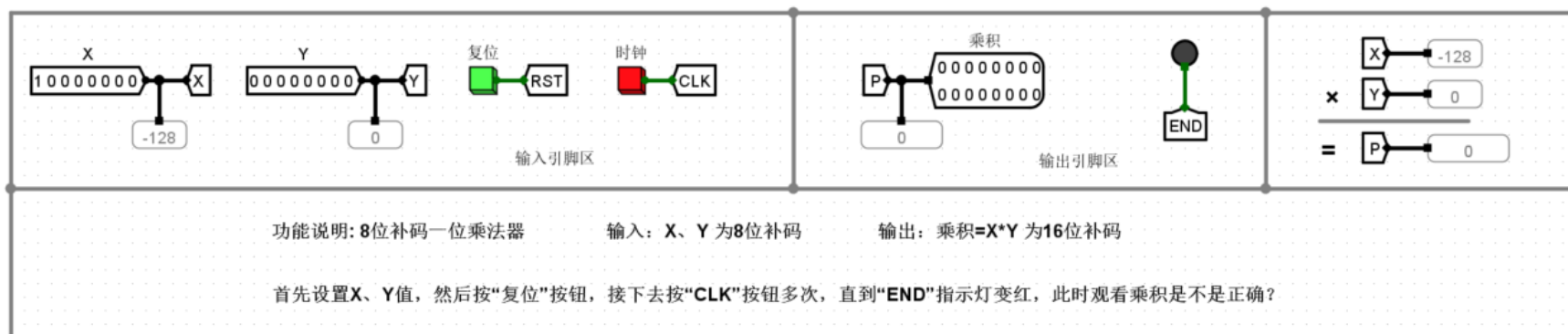
• 另一种附加电路（复杂版）：

- 第一步：计算XZ、YZ、XYZ
 - 如果X=0，则XZ=0；否则，XZ=1
 - 如果Y=0，则YZ=0；否则，YZ=1
 - 如果X=0且Y=0，或者X≠0且Y≠0，则XYZ=0；如果X=0且Y≠0，或者X≠0且Y=0，则XYZ=1
- 第二步：获取X、Y的符号位（SX、SY），以及SP
 - 如果X、Y都是正数（包括0），或都是负数，则SP=0；否则，SP=1
- 第三步：计算PX（16位）、PY（16位）
 - PX = SX（1位） X（8位） 0000000（7位）
 - PY = SY（1位） Y（8位） 0000000（7位）
- 第四步：计算PX*、PY*
 - $PX^* = SP$ PX的补数（取15位） 即： $PX^* = (-128) * X$
 - $PY^* = SP$ PY的补数（取15位） 即： $PY^* = (-128) * Y$
- 第五步：计算PS（80H=-128）
 - 如果X≠-128 且 Y≠-128，则PS=00
 - 如果X=-128 且 Y≠-128，则PS=01
 - 如果X≠-128 且 Y=-128，则PS=10
 - 如果X=-128 且 Y=-128，则PS=11
- 第六步：计算P（最终的乘积）（4000H=16384）
 - 如果PS=00且XYZ=0，则P=P* 正常情况， $P = P^*$
 - 如果PS=01且XYZ=0，则P=PY* $X = -128$ ， $P = (-128) * Y$
 - 如果PS=10且XYZ=0，则P=PX* $Y = -128$ ， $P = (-128) * X$
 - 如果PS=11且XYZ=0，则P=4000H $X = -128$ ， $Y = -128$ ， $P = (-128) * (-128) = 16384$
 - 如果XYZ=1，则P=0000H $X = 0$ 且 $Y \neq 0$ ，或者 $X \neq 0$ 且 $Y = 0$ ， $P = 0$



- 8位补码一位乘法器的运行：

- (1) 设置X和Y的值
- (2) 鼠标点击电路图中的“复位”
- (3) 鼠标点击电路图中的“时钟”若干次，直到运算结束指示灯（END）变红
- (4) 观看运算结果（P）是否正确

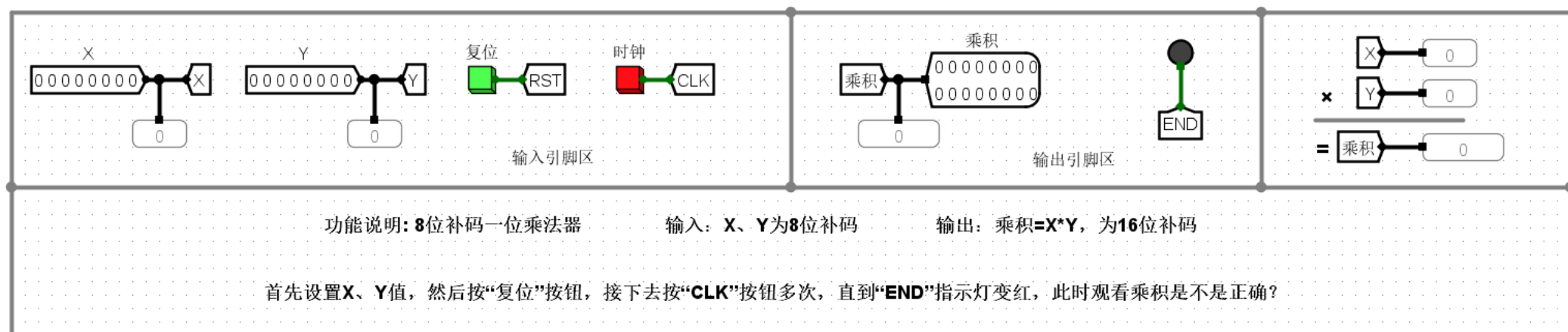


- 请同学们设置不同的X、Y值（补码）

- X=-128, Y=其他, P是否正确?
- X=其他, Y=-128, P是否正确?
- X=0, Y=非0, P是否=0?
- X=非0, Y=0, P是否=0?
- X=-128, Y=-128, P是否=16384?
- X=127, Y=127, P是否=16129?
- X=127, Y=-128, P是否=-16256?
- X=-128, Y=127, P是否=-16256?
- X=其他, Y=其他, P是否正确?

• 4、8位补码一位乘法器（用8位无符号数一位乘法器实现）（挑战性实验）

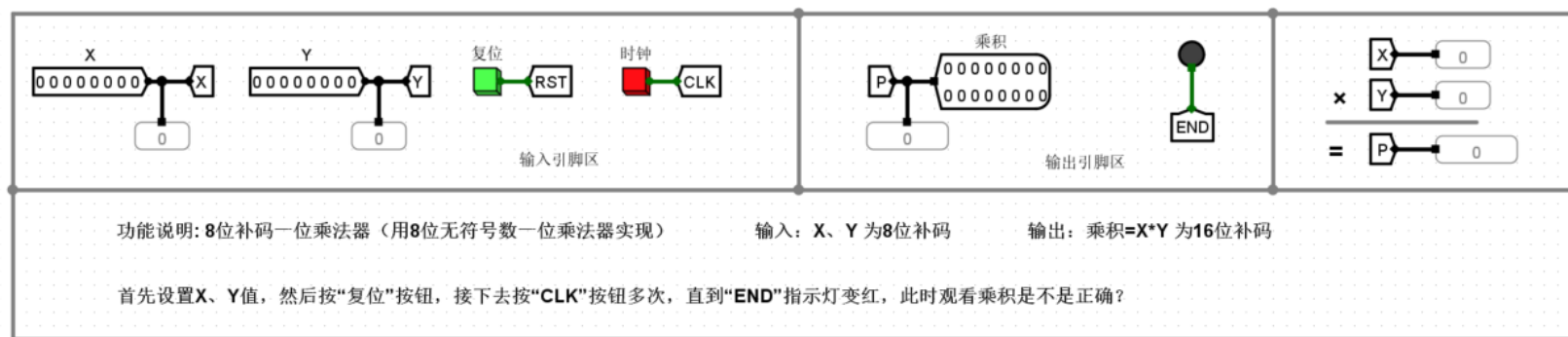
- 输入：X（8位，补码）、Y（8位，补码）、复位（RST）、时钟（CLK）
- 输出：乘积（16位，补码）、运算结束指示灯（END）



请同学们在此处设计电路！

8位补码一位乘法器（用8位无符号数一位乘法器实现）

- 8位补码一位乘法器（用8位无符号一位乘法器实现）的运行：
 - (1) 设置X和Y的值
 - (2) 鼠标点击电路图中的“复位”
 - (3) 鼠标点击电路图中的“时钟”若干次，直到运算结束指示灯（END）变红
 - (4) 观看运算结果（P）是否正确



- 请同学们设置不同的X、Y值（补码）
 - X=-128, Y=其他, P是否正确?
 - X=其他, Y=-128, P是否正确?
 - X=0, Y=非0, P是否=0?
 - X=非0, Y=0, P是否=0?
 - X=-128, Y=-128, P是否=16384?
 - X=127, Y=127, P是否=16129?
 - X=127, Y=-128, P是否=-16256?
 - X=-128, Y=127, P是否=-16256?
 - X=其他, Y=其他, P是否正确?

需要同学们完成的设计实验的内容

- ① 16位快速加法器（组内并行、组间并行）（设计实验）——文件名：16位快速加法器（组内并行、组间并行）.circ
- ② 32位快速加法器（组内并行、组间串行）（设计实验）——文件名：32位快速加法器（组内并行、组间串行）.circ
- ③ 6位原码阵列乘法器（设计实验）——文件名：6位原码阵列乘法器.circ
- ④ 6位补码阵列乘法器（设计实验）——文件名：6位补码阵列乘法器.circ
- ⑤ 8位原码一位乘法器（设计实验）——文件名：8位原码一位乘法器.circ
- ⑥ 8位补码一位乘法器（采用8位无符号一位乘法器实现）（挑战性实验）——文件名：8位补码一位乘法器（采用8位无符号一位乘法器实现）.circ

- 完成挑战性实验，本次实验成绩加10分

实验报告文件提交要求

- 1、请按照FTP上的实验报告模板（**厦门大学计算机组成原理实验报告样本（第2次实验）.docx**），撰写实验报告（Word版本），实验报告命名为：**学号+姓名+第2次实验报告**
- 2、请将实验报告和6个设计文件打包为1个压缩文件，压缩文件命名为：**学号+姓名+第2次实验.zip**
- 3、将压缩文件上传到FTP上，第2次实验报告提交**截止时间**（2周内）：2023年4月5日晚上24点

Thanks