

计算机组成与嵌入式系统

--运算器实验

老师：徐文辉

QQ: 4127164

电话：18202799815

Email: xuwenhui@hust.edu.cn

实验环境

- Logisim仿真软件

实验内容

- 快速加法器
- 乘法器
- 算术逻辑运算单元

| alu* | |
|-------------------------------------|--------------|
| <input type="checkbox"/> | ☆8位可控加减法器 |
| <input type="checkbox"/> | ☆4位先行进位74182 |
| <input type="checkbox"/> | ☆4位快速加法器 |
| <input type="checkbox"/> | ☆16位快速加法器 |
| <input type="checkbox"/> | ☆32位快速加法器 |
| <input type="checkbox"/> | ☆5位阵列乘法器 |
| <input type="checkbox"/> | ☆6位补码阵列乘法器 |
| <input checked="" type="checkbox"/> | ☆5位无符号乘法流水线 |
| <input type="checkbox"/> | ☆原码一位乘法器 |
| <input type="checkbox"/> | ☆补码一位乘法器 |
| <input type="checkbox"/> | ☆算术逻辑运算单元ALU |
| <input type="checkbox"/> | ★ALU自动测试 |
| <input type="checkbox"/> | ★5位乘法与门阵列 |
| <input type="checkbox"/> | ★乘法线流水接口 |
| <input type="checkbox"/> | ★一位全加器FA封装1 |
| <input type="checkbox"/> | ★一位全加器FA封装2 |
| <input type="checkbox"/> | ★一位全加器FA封装3 |
| <input type="checkbox"/> | ★8位串行加法器 |

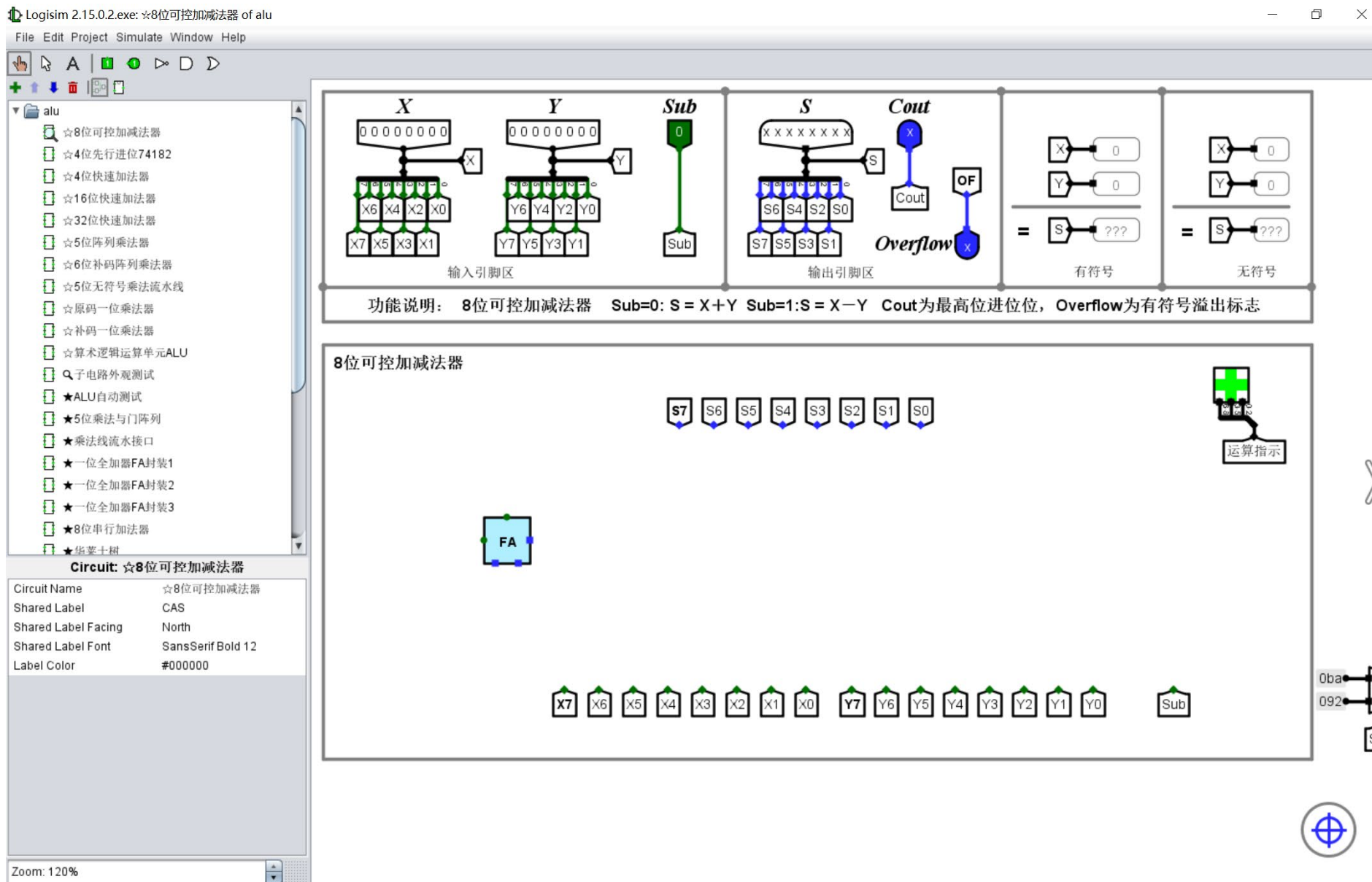
实验一：快速加法器实验

快速加法器实验

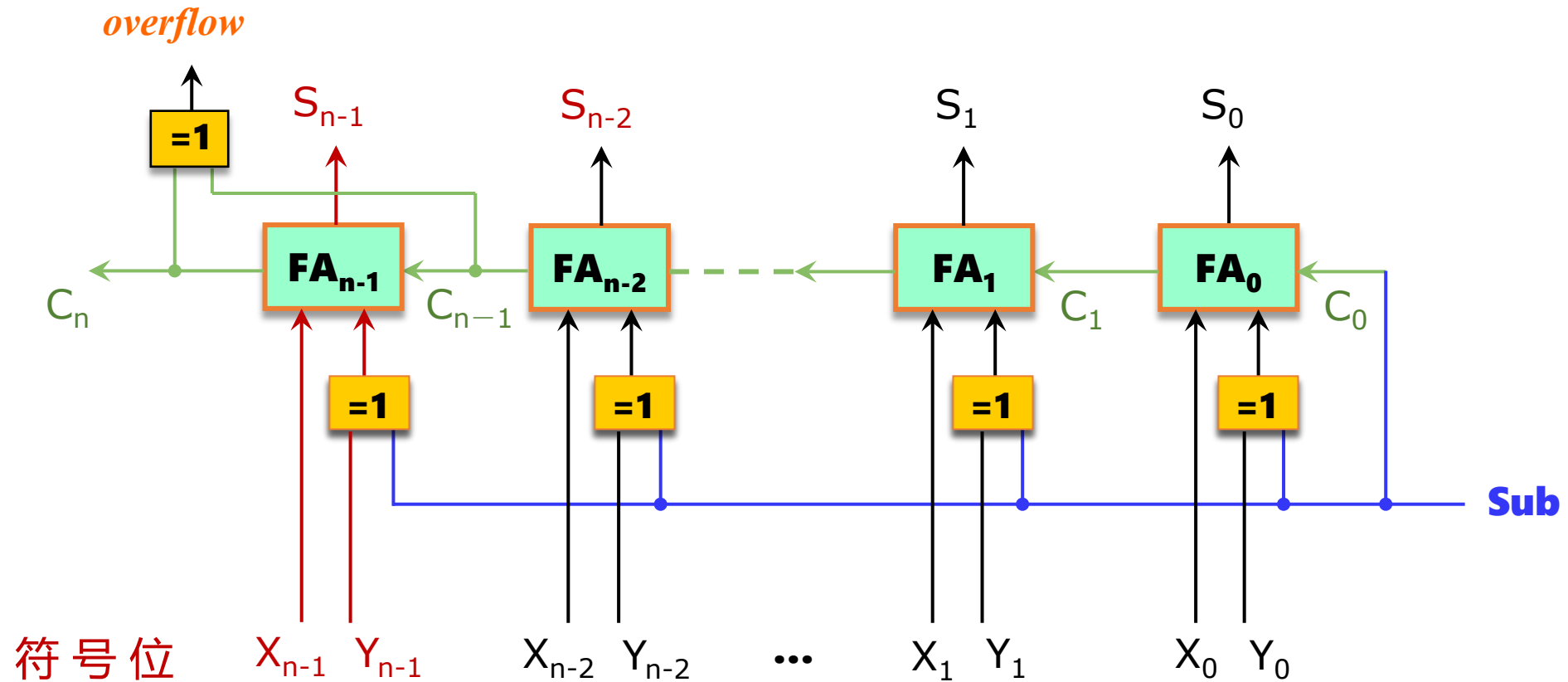
• 实验目的

- 掌握串行加法器逻辑实现
 - 能设计8位可控加减法电路
- 掌握快速加法器逻辑实现
 - 能设计4位先行进位电路
 - 能设计4位快速加法器
- 掌握组内先行、组间先行的基本原理
 - 利用4位快速加法器构建16位、32位快速加法器
 - 能分析相关电路延迟

快速加法器实验----8位可控加减法器设计



快速加法器实验----8位可控加减法器设计



减法的避免减少了逻辑器件，控制信号Sub如何产生？

快速加法器实验-----4位先行进位电路设计

- 并行加法器进位链

$$S_i = X_i \oplus Y_i \oplus C_{i-1}$$

$$C_i = \underline{X_i Y_i} + (\underline{X_i \oplus Y_i}) C_{i-1}$$

$$G_i = X_i Y_i \quad \text{进位生成函数} \quad \text{Generate}$$

$$P_i = X_i \oplus Y_i \quad \text{进位传递函数} \quad \text{Propagate}$$

$$C_i = G_i + P_i C_{i-1}$$

快速加法器实验-----4位先行进位电路设计

- 并行加法器进位链

$$C_1 = \underline{G_1 + P_1 C_0}$$

$$C_2 = G_2 + P_2 \underline{C_1}$$

$$= G_2 + P_2 (G_1 + P_1 C_0) = G_2 + P_2 G_1 + P_2 P_1 C_0$$

$$C_3 = G_3 + P_3 C_2$$

$$= G_3 + P_3 (G_2 + P_2 G_1 + P_2 P_1 C_0)$$

$$= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_0$$

快速加法器实验-----4位先行进位电路设计

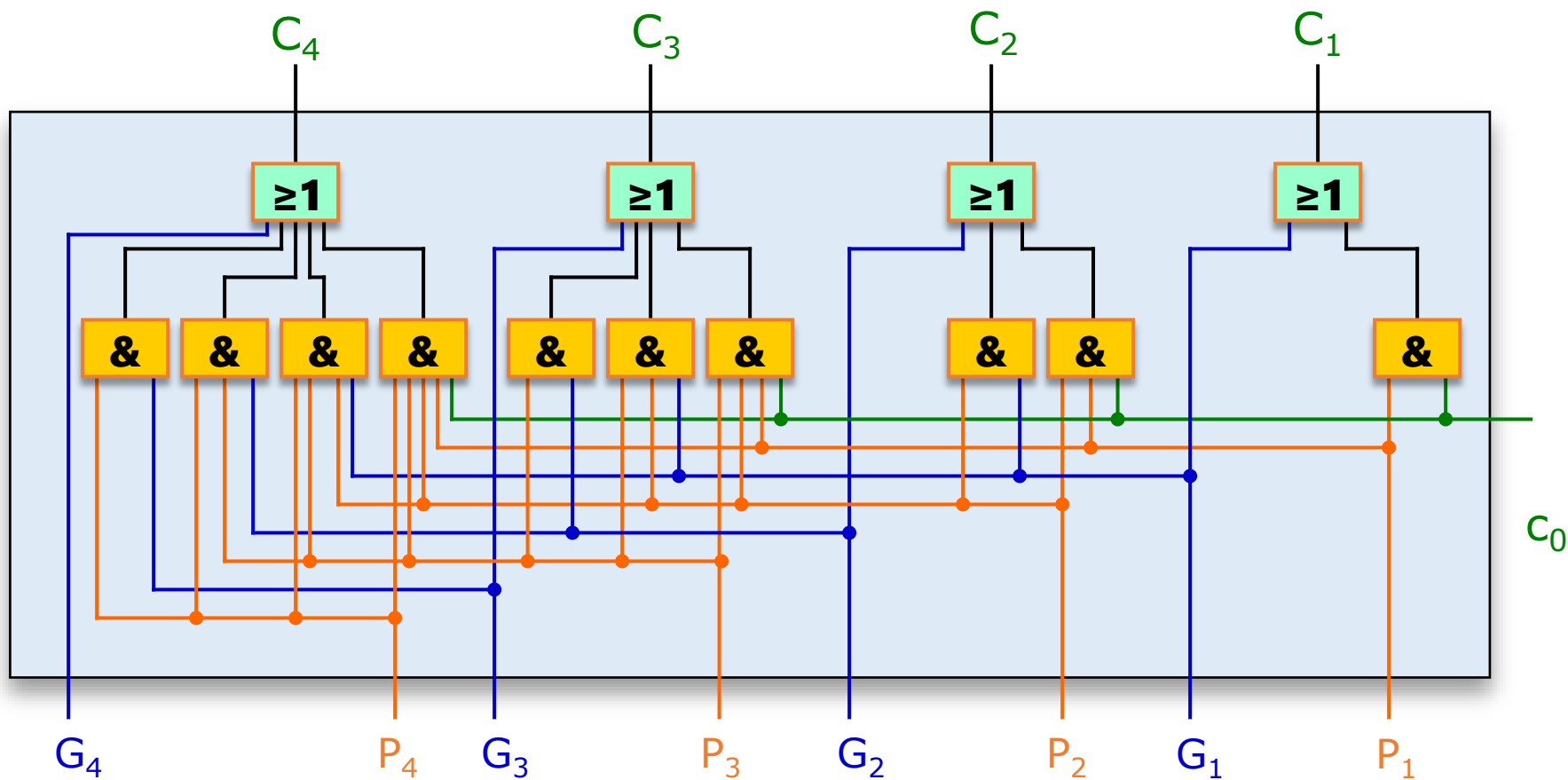
- 并行加法器进位链

$$C_n = G_n + P_n G_{n-1} + P_n P_{n-1} G_{n-2} + P_n P_{n-1} P_{n-2} G_{n-3} \dots + P_n P_{n-1} \dots P_1 C_0$$

- 进位输出仅与最低位进位输入 C_0 有关
- 位数越长，进位链电路复杂度越高
- 通常按照4位一组进行分组运算

$$C_4 = G_4 + P_4 G_3 + P_4 P_3 G_2 + P_4 P_3 P_2 G_1 + P_4 P_3 P_2 P_1 C_0$$

快速加法器实验-----4位先行进位电路设计



2级门电路延迟

快速加法器实验-----4位先行进位电路设计

Logisim 2.15.0.2.exe: ☆4位先行进位74182 of alu

File Edit Project Simulate Window Help

alul

- ☆8位可控加减法器
- ☆4位先行进位74182
- ☆4位快速加法器
- ☆16位快速加法器
- ☆32位快速加法器
- ☆5位阵列乘法器
- ☆6位补码阵列乘法器
- ☆5位无符号乘法流水线
- ☆原码一位乘法器
- ☆补码一位乘法器
- ☆算术逻辑运算单元ALU
- Q子电路外观测试
- ★ALU自动测试
- ★5位乘法与门阵列
- ★乘法流水线接口
- ★一位全加器FA封装1
- ★一位全加器FA封装2
- ★一位全加器FA封装3
- ★8位串行加法器
- ★华莱士树

Circuit: ☆4位先行进位74182

| | |
|---------------------|-------------------|
| Circuit Name | ☆4位先行进位74182 |
| Shared Label | CLA74182 |
| Shared Label Facing | North |
| Shared Label Font | SansSerif Bold 12 |
| Label Color | #000000 |

Zoom: 110%

输入引脚区

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| G^4 | P^4 | G^3 | P^3 | G^2 | P^2 | G^1 | P^1 | Cin |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G4 | P4 | G3 | P3 | G2 | P2 | G1 | P1 | Cin |

输出引脚区

| | | | | | |
|-------|-------|-------|-------|-------|-------|
| C^4 | C^3 | C^2 | C^1 | G^* | P^* |
| x | x | x | x | x | x |
| C4 | C3 | C2 | C1 | G* | P* |

请根据以上引脚以及隧道信号设计完成74LS182先行进位电路，尽可能使用多输入逻辑门，并分析电路延迟

4位先行进位电路

P^* G^* C^4 C^3 C^2 C^1

G4 P4 G3 P3 G2 P2 G1 P1 Cin

快速加法器实验-----4位快速加法器设计

Logisim 2.15.0.2.exe: ☆4位快速加法器 of alu

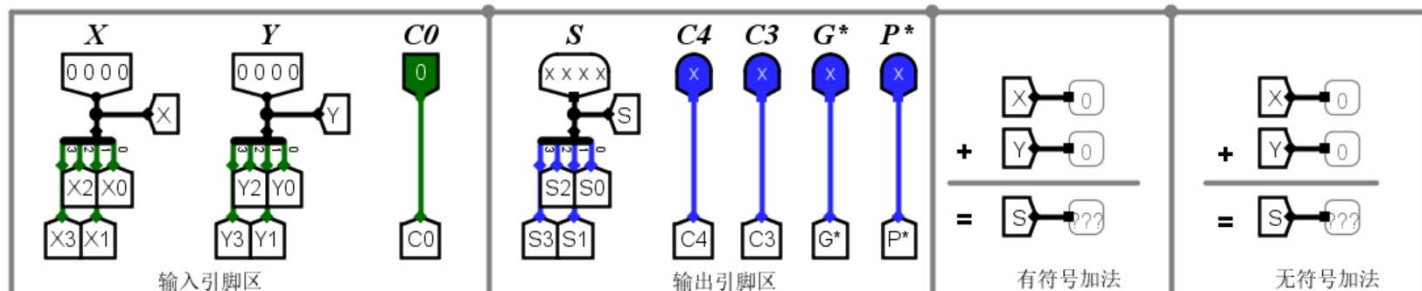
File Edit Project Simulate Window Help



Circuit: ☆4位快速加法器

Circuit Name ☆4位快速加法器
Shared Label 4位快速加法器
Shared Label Facing North
Shared Label Font SansSerif Bold 12
Label Color #000000

Zoom: 130%



功能说明:4位Adder $S=X+Y$, C4、C3分别为最高位、次高位进位位, C0为进位输入, $G^* P^*$ 为成组进位生成传递函数

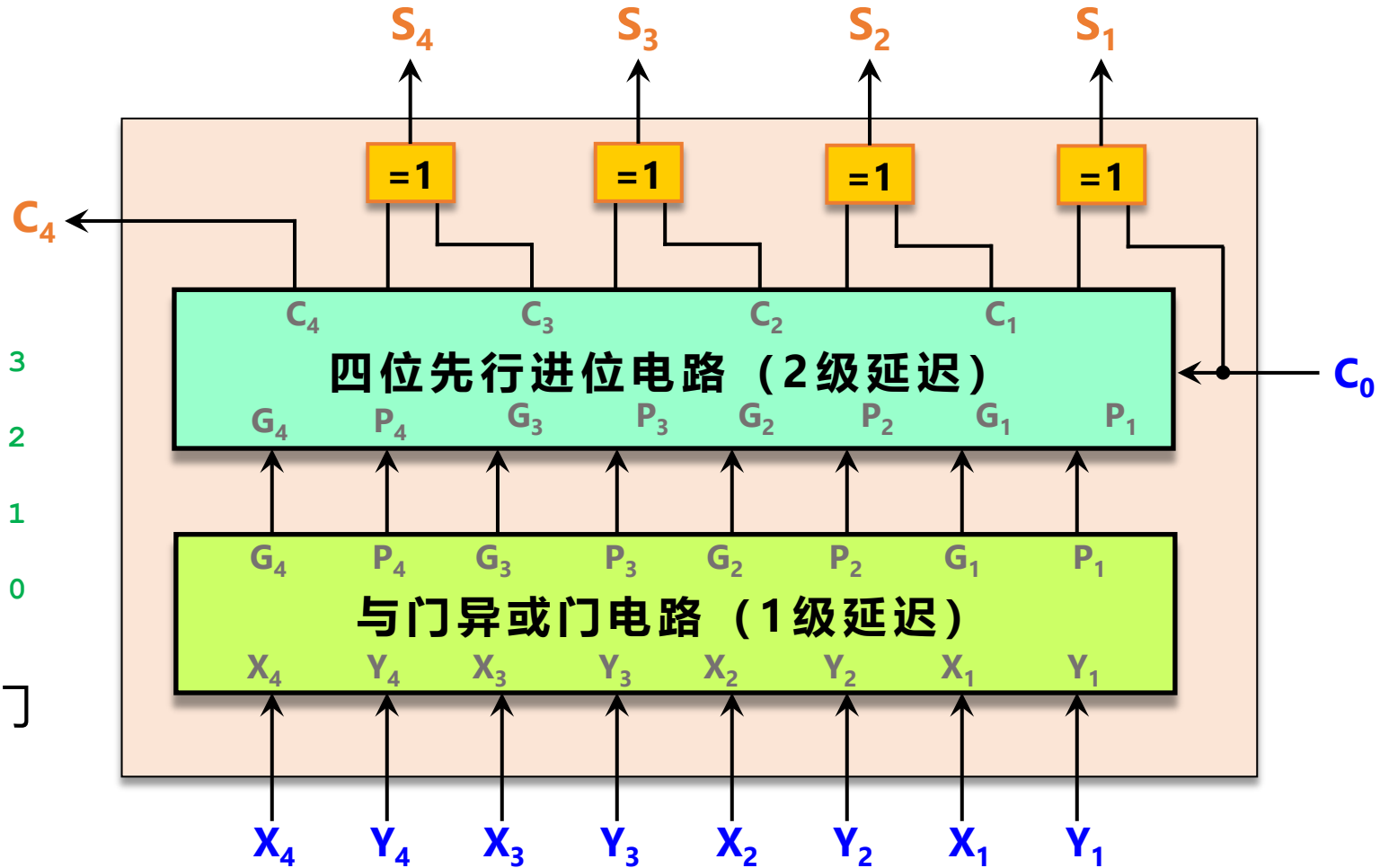
4位快速加法器



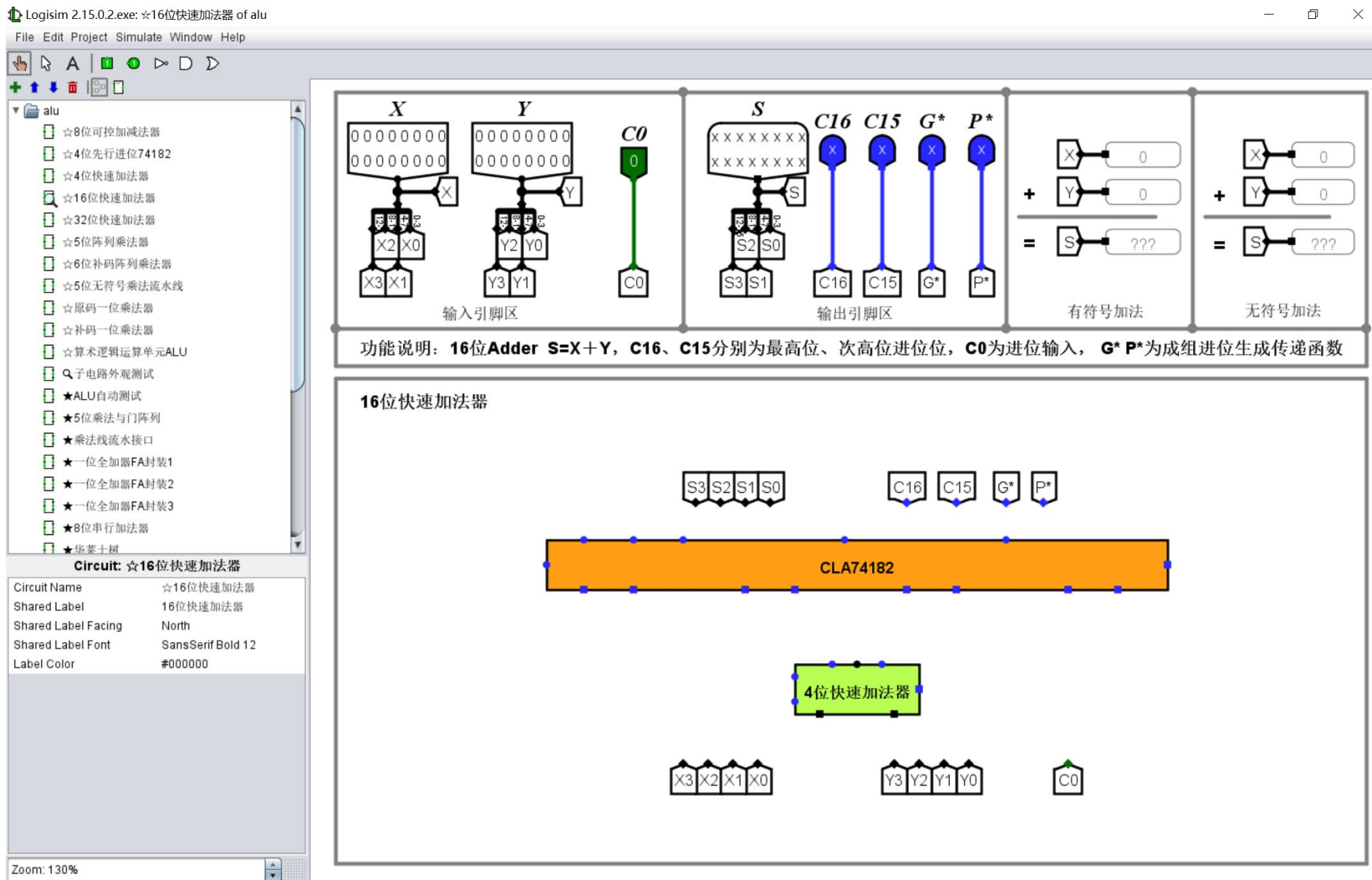
快速加法器实验-----4位快速加法器设计

- $S_4 = X_4 \oplus Y_4 \oplus C_3 = P_4 \oplus C_3$
- $S_3 = X_3 \oplus Y_3 \oplus C_2 = P_3 \oplus C_2$
- $S_2 = X_2 \oplus Y_2 \oplus C_1 = P_2 \oplus C_1$
- $S_1 = X_1 \oplus Y_1 \oplus C_0 = P_1 \oplus C_0$

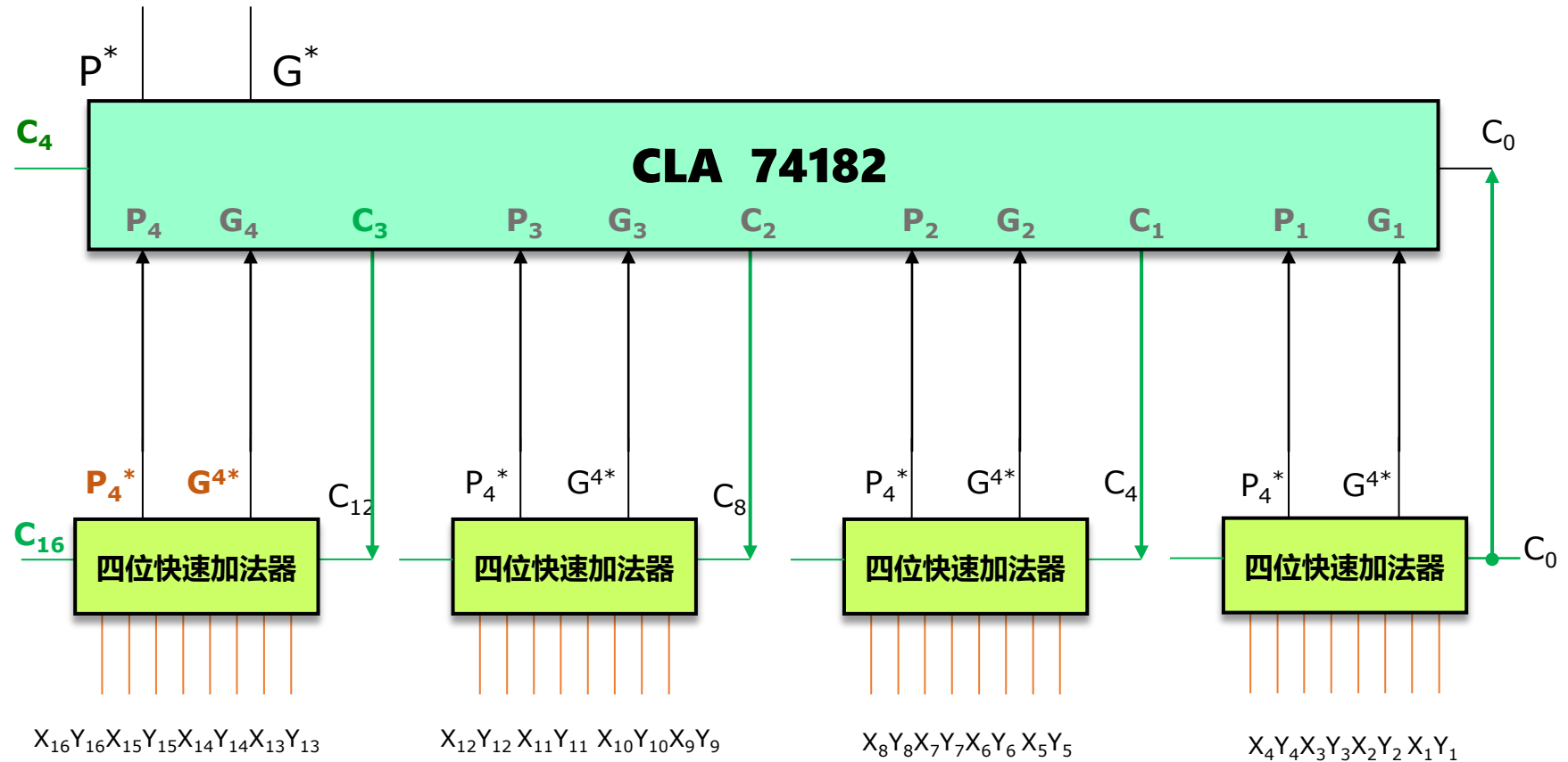
- 进位信号得到后，求和只需一级异或门即可完成



快速加法器实验-----16位快速加法器设计



快速加法器实验-----16位快速加法器设计



①生成 P^* , G^* 需3T → ②生成 C_3/C_{12} 需2T → ③求和需3T

快速加法器实验-----32位快速加法器设计

Logisim 2.15.0.2.exe: ☆32位快速加法器 of alu

File Edit Project Simulate Window Help

Input: X, Y, C0

Output: S, C32, C31

有符号加法: $X + Y = S$

无符号加法: $X + Y = S$

功能说明: 利用已完成的74182以及4位快速加法器构建速度最快的32位加法器 $S = X + Y$, 并分析电路延迟: C32、C31分别为最高位、次高位进位位, C0为进位输入

32位快速加法器

CLA74182

CLA74182

4位快速加法器

4位快速加法器

4位快速加法器

Input: X7, X6, X5, X4, X3, X2, X1, X0, Y7, Y6, Y5, Y4, Y3, Y2, Y1, Y0, C0

Output: S7, S6, S5, S4, S3, S2, S1, S0

Circuit: ☆32位快速加法器

Circuit Name: ☆32位快速加法器

Shared Label: 32位加法器

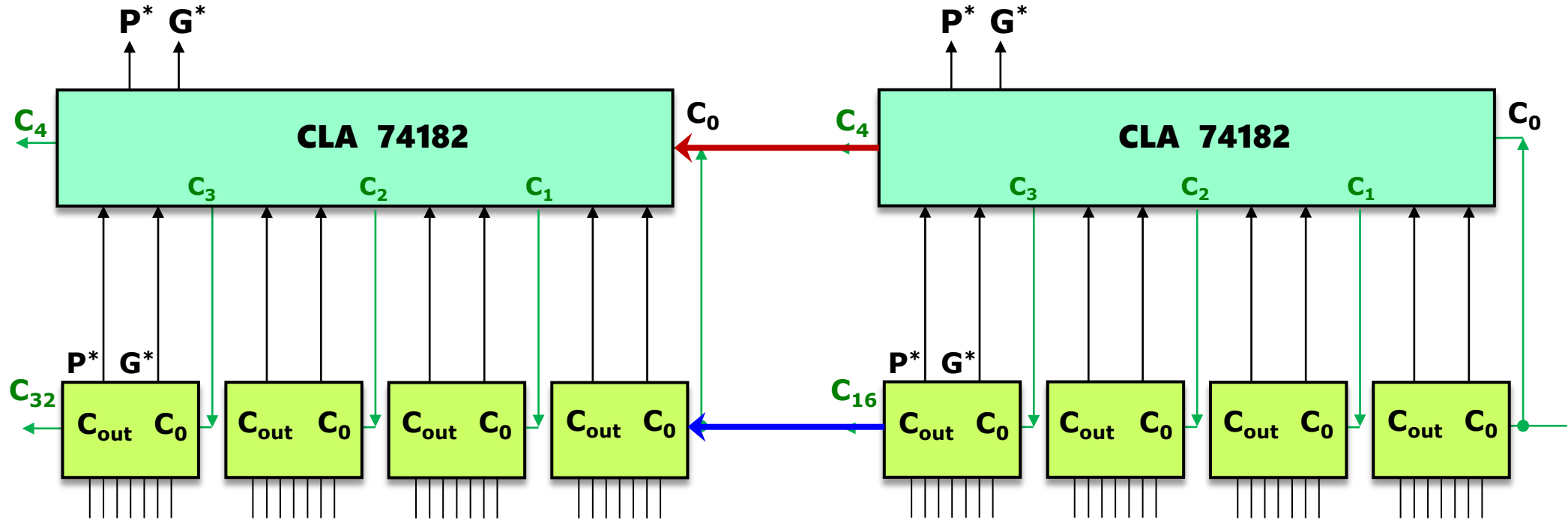
Shared Label Facing: North

Shared Label Font: SansSerif Plain 12

Label Color: #000000

Zoom: 87%

快速加法器实验-----32位快速加法器设计



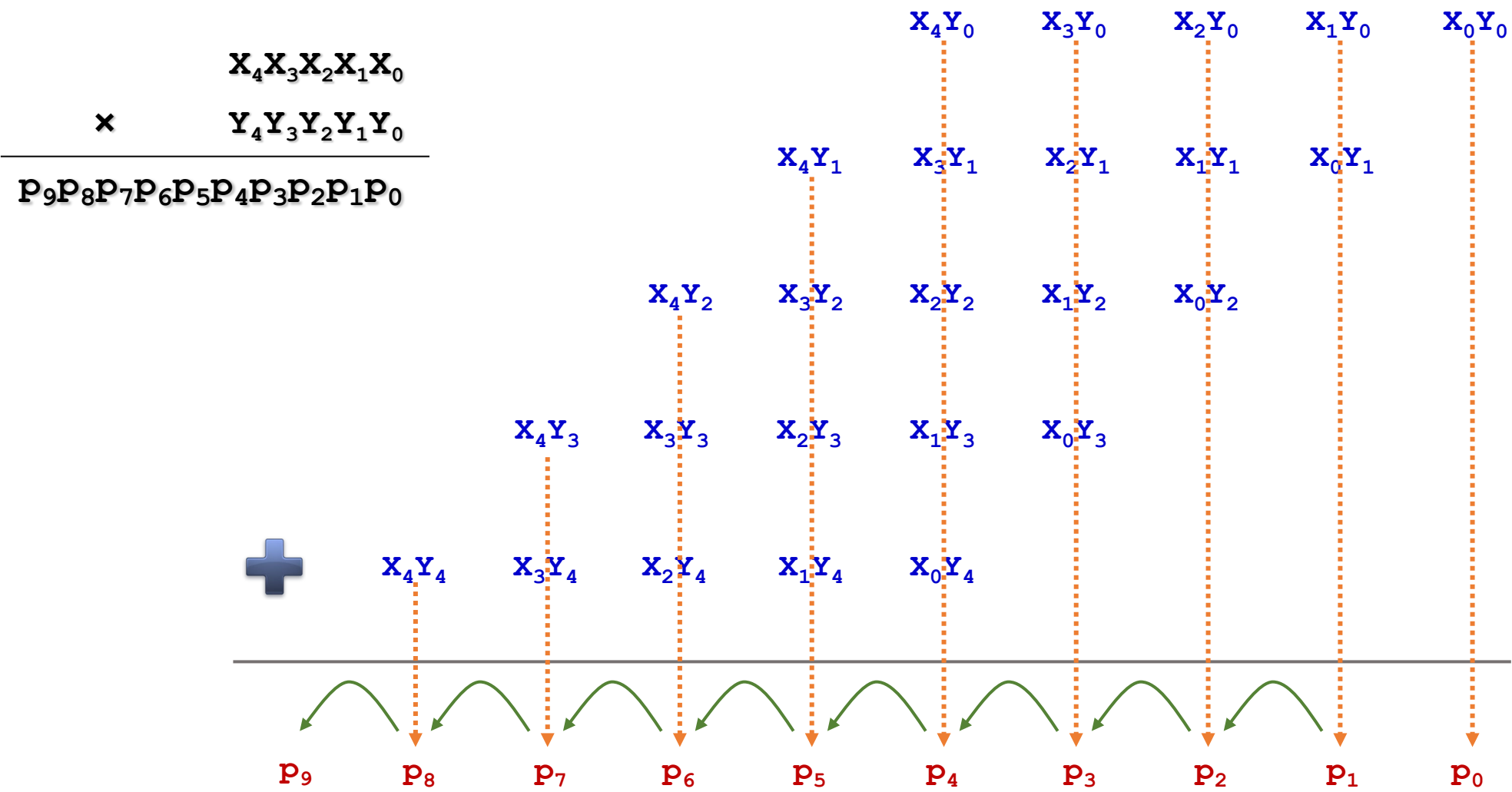
实验二：乘法器实验

乘法器实验

• 实验目的

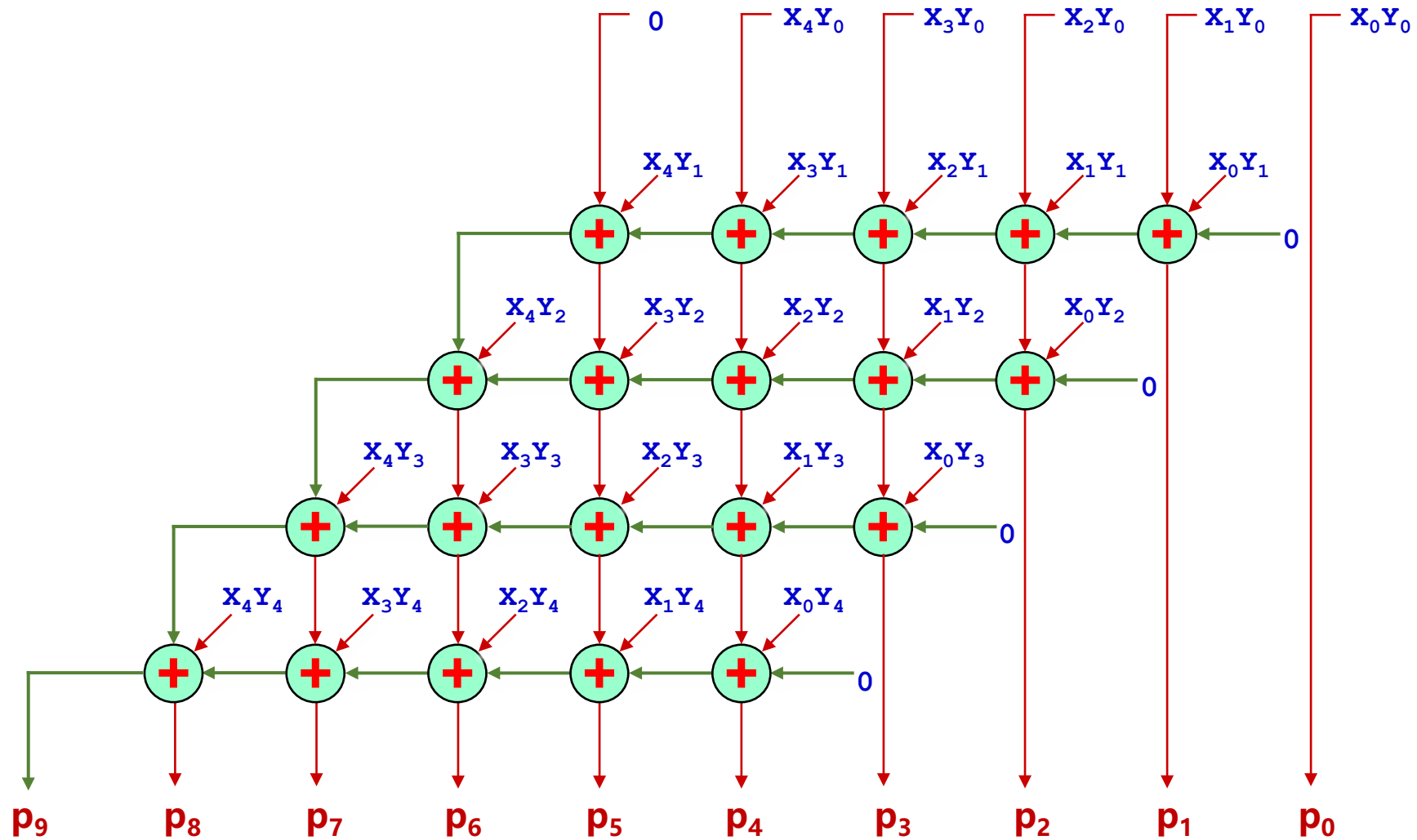
- 理解阵列乘法器的实现原理
 - 能设计无符号阵列乘法器电路
 - 能设计有符号补码阵列乘法器电路
- 掌握原码、补码一位乘法基本原理
 - 能设计原码、补码一位乘法器
 - 重点掌握寄存器、多路选择器的使用
 - 能设计简单的状态机进行数据通路控制

乘法器实验-----5位无符号数阵列乘法器

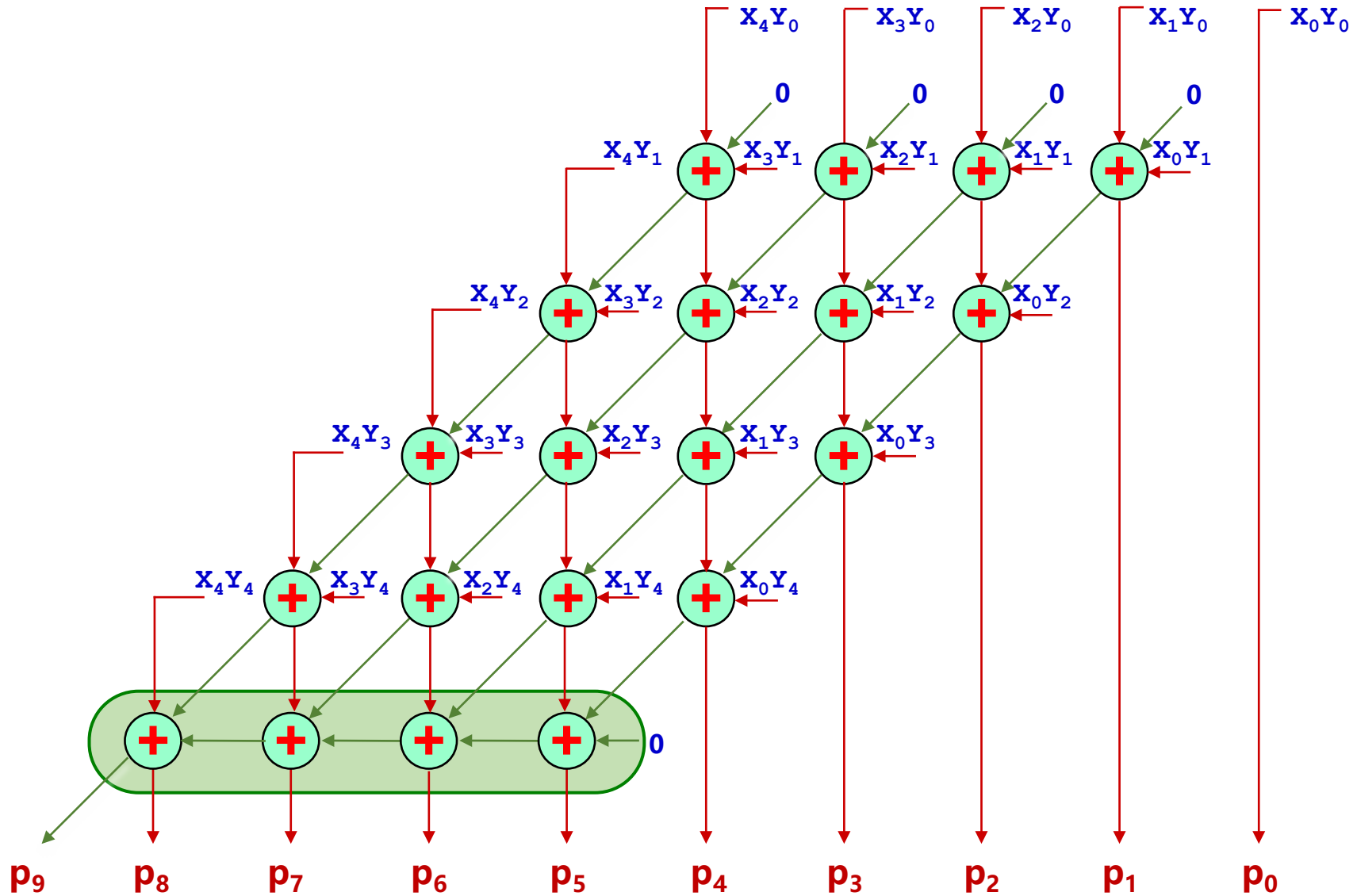


先计算相加数，然后逐列相加

乘法器实验-----5位无符号数阵列乘法器



乘法器实验-----5位无符号数阵列乘法器



乘法器实验-----5位无符号数阵列乘法器

Logisim 2.15.0.2.exe: ☆5位阵列乘法器 of alu

File Edit Project Simulate Window Help

alul

- ☆8位可控加减法器
- ☆4位先行进位74182
- ☆4位快速加法器
- ☆16位快速加法器
- ☆32位快速加法器
- ☆5位阵列乘法器
- ☆6位补码阵列乘法器
- ☆5位无符号乘法流水线
- ☆原码一位乘法器
- ☆补码一位乘法器
- ☆算术逻辑运算单元ALU
- Q子电路外观测试
- ★ALU自动测试
- ★5位乘法与门阵列
- ★乘法流水线接口
- ★一位全加器FA封装1
- ★一位全加器FA封装2
- ★一位全加器FA封装3
- ★8位串行加法器
- ★华莱士树

Circuit: ☆5位阵列乘法器

Circuit Name: ☆5位阵列乘法器

Shared Label: 5位阵列乘法器

Shared Label Facing: North

Shared Label Font: SansSerif Plain 12

Label Color: #000000

Zoom: 110%

输入引脚区

输出引脚区

乘积

功能说明: 利用一位全加器构成5位×5位的无符号数阵列乘法器, 乘积=X×Y, 并分析电路延迟:

5位无符号阵列乘法器

X4Y0 X3Y0 X2Y0 X1Y0 X0Y0

X4Y1 X3Y1 X2Y1 X1Y1 X0Y1

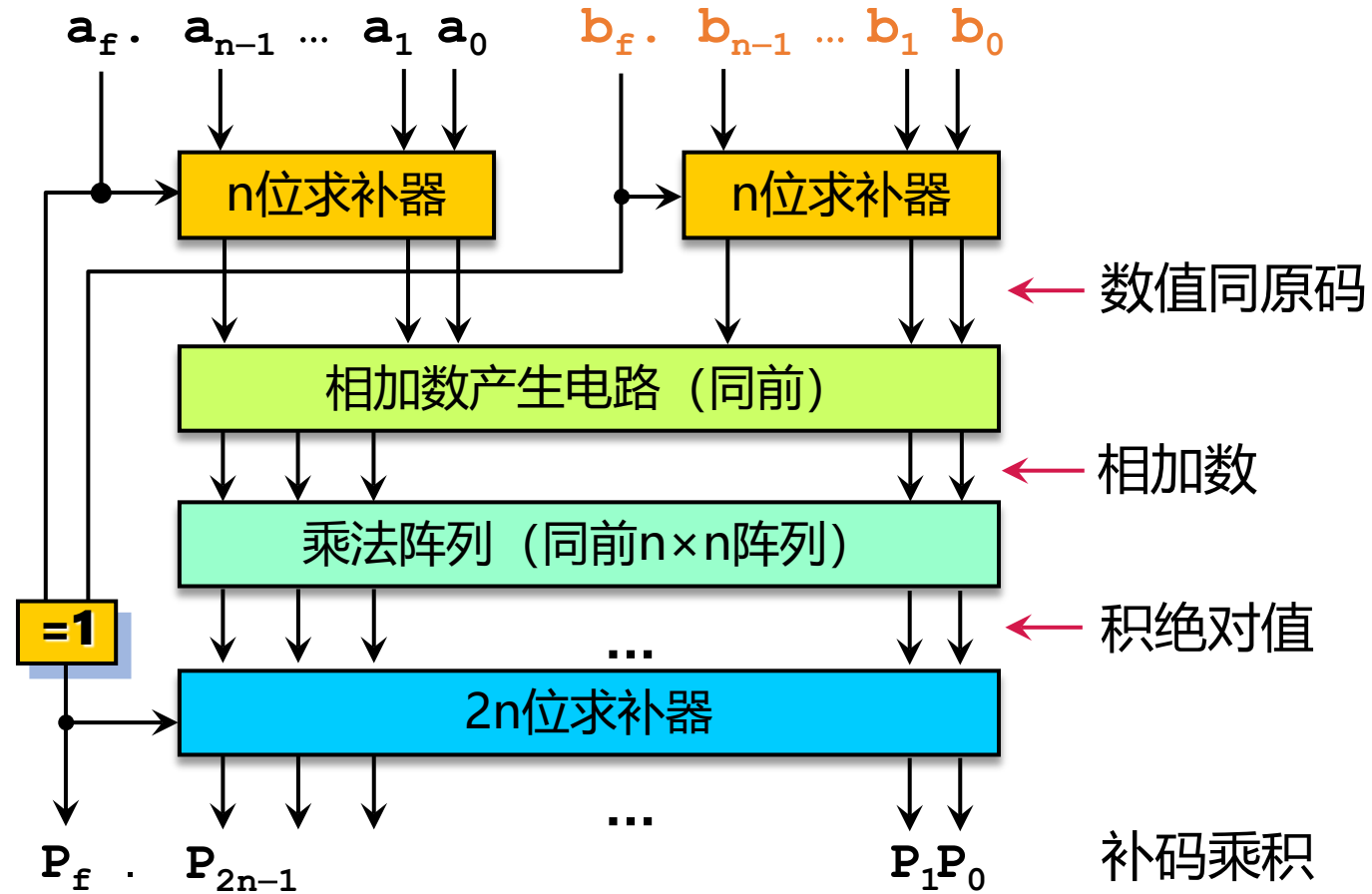
X4Y2 X3Y2 X2Y2 X1Y2 X0Y2

X4Y3 X3Y3 X2Y3 X1Y3 X0Y3

X4Y4 X3Y4 X2Y4 X1Y4 X0Y4

P9 P8 P7 P6 P5 P4 P3 P2 P1 P0

乘法器实验-----6位补码阵列乘法器



乘法器实验-----6位补码阵列乘法器

Logisim 2.15.0.2.exe: ☆6位补码阵列乘法器 of alu

File Edit Project Simulate Window Help

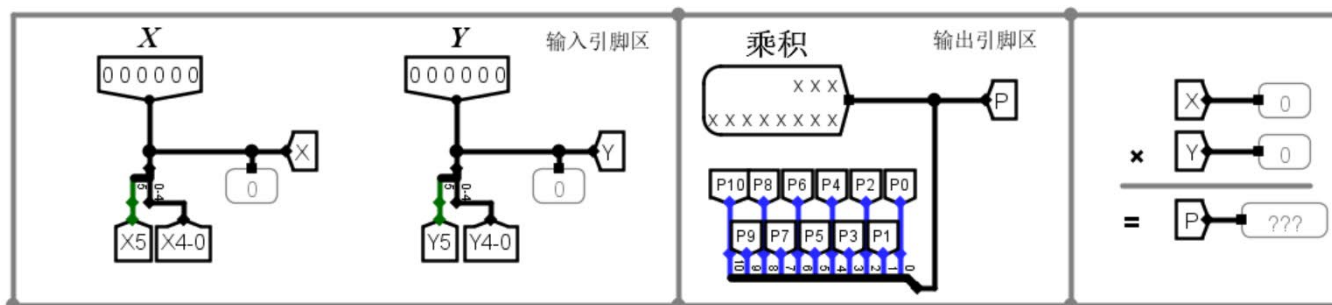


- ☆8位可控加减法器
- ☆4位先行进位74182
- ☆4位快速加法器
- ☆16位快速加法器
- ☆32位快速加法器
- ☆5位阵列乘法器
- ☆6位补码阵列乘法器
- ☆5位无符号乘法流水线
- ☆原码一位乘法器
- ☆补码一位乘法器
- ☆算术逻辑运算单元ALU
- Q子电路外观测试
- ★ALU自动测试
- ★5位乘法与门阵列
- ★乘法流水线接口
- ★一位全加器FA封装1
- ★一位全加器FA封装2
- ★一位全加器FA封装3
- ★8位串行加法器
- ★华莱士树

Circuit: ☆6位补码阵列乘法器

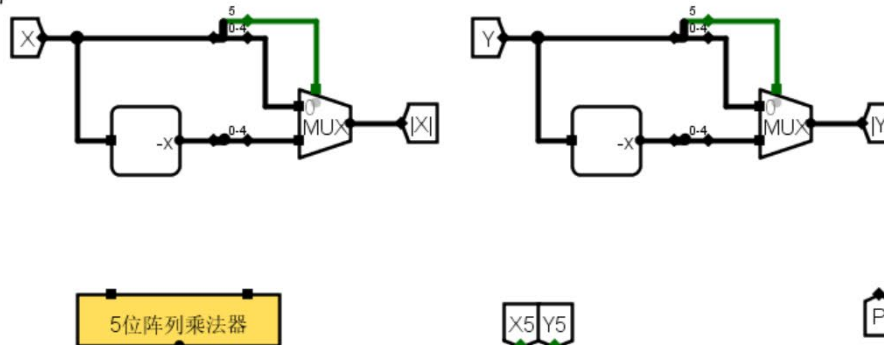
Circuit Name ☆6位补码阵列乘法器
Shared Label 6位补码乘法器
Shared Label Facing North
Shared Label Font SansSerif Plain 12
Label Color #000000

Zoom: 130%

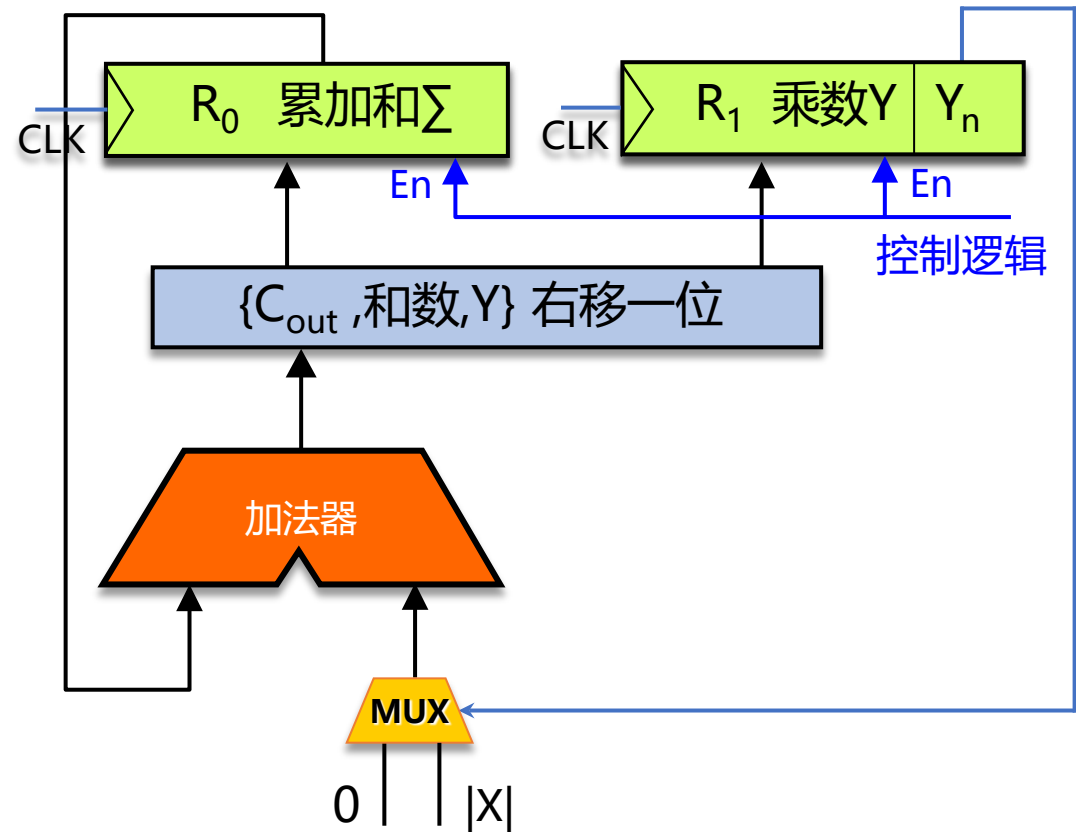
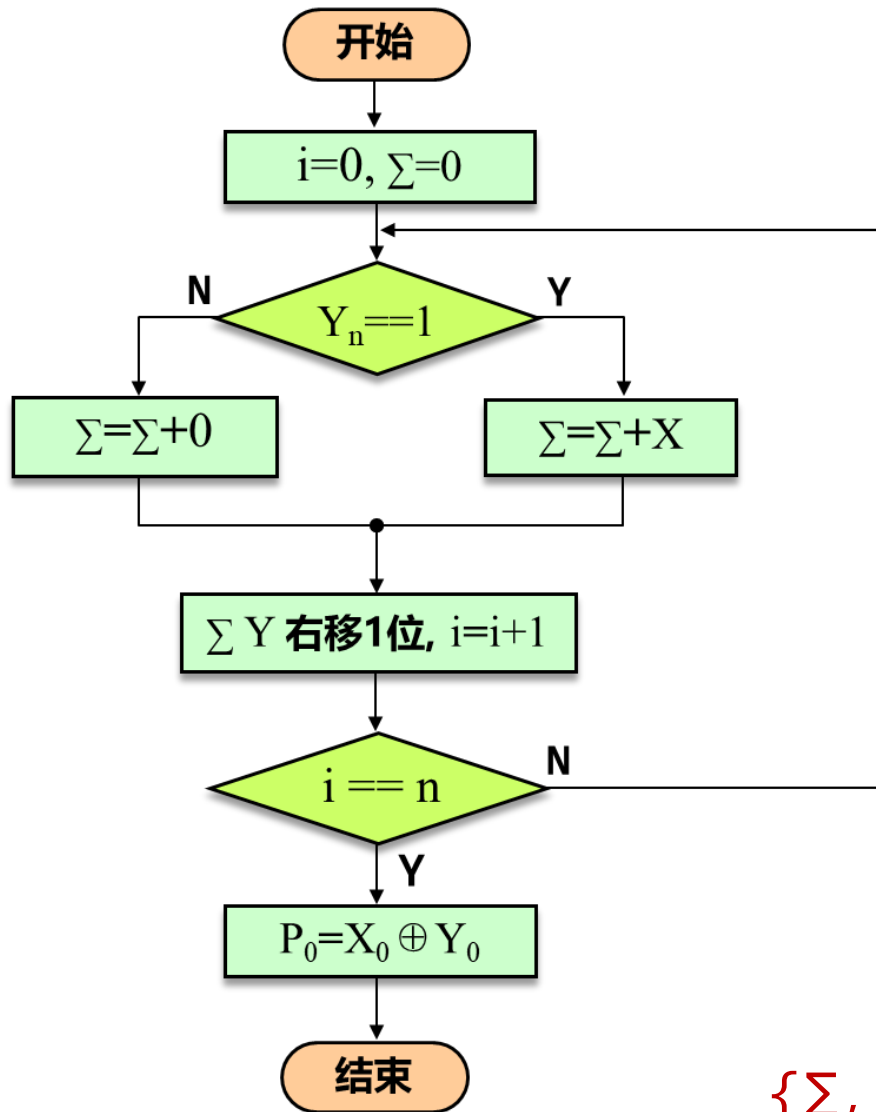


功能说明：利用五位阵列乘法器6位×6位的补码阵列乘法器，乘积 $P = X \times Y$ ，尝试计算 -32×1 ，分析结果

6位补码阵列乘法器

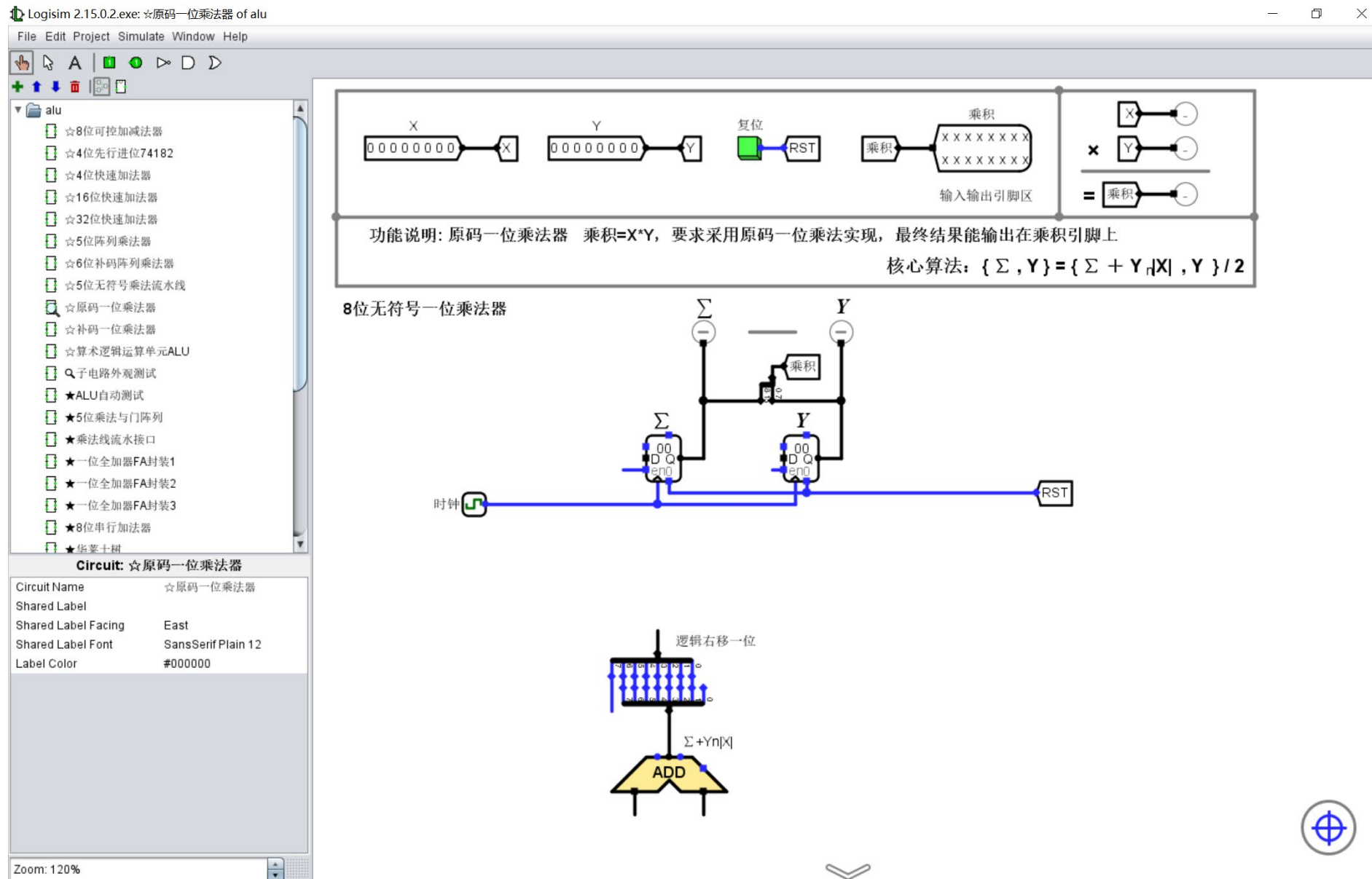


乘法器实验-----原码一位乘法器

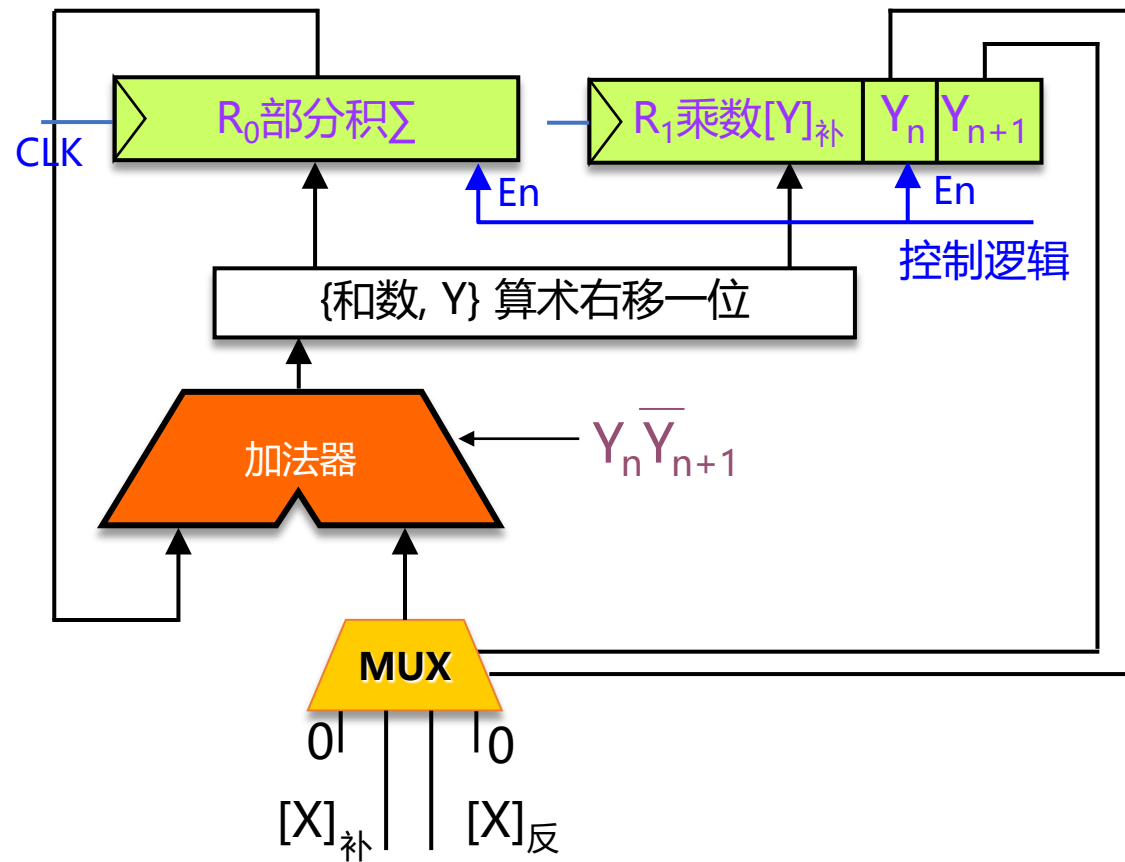
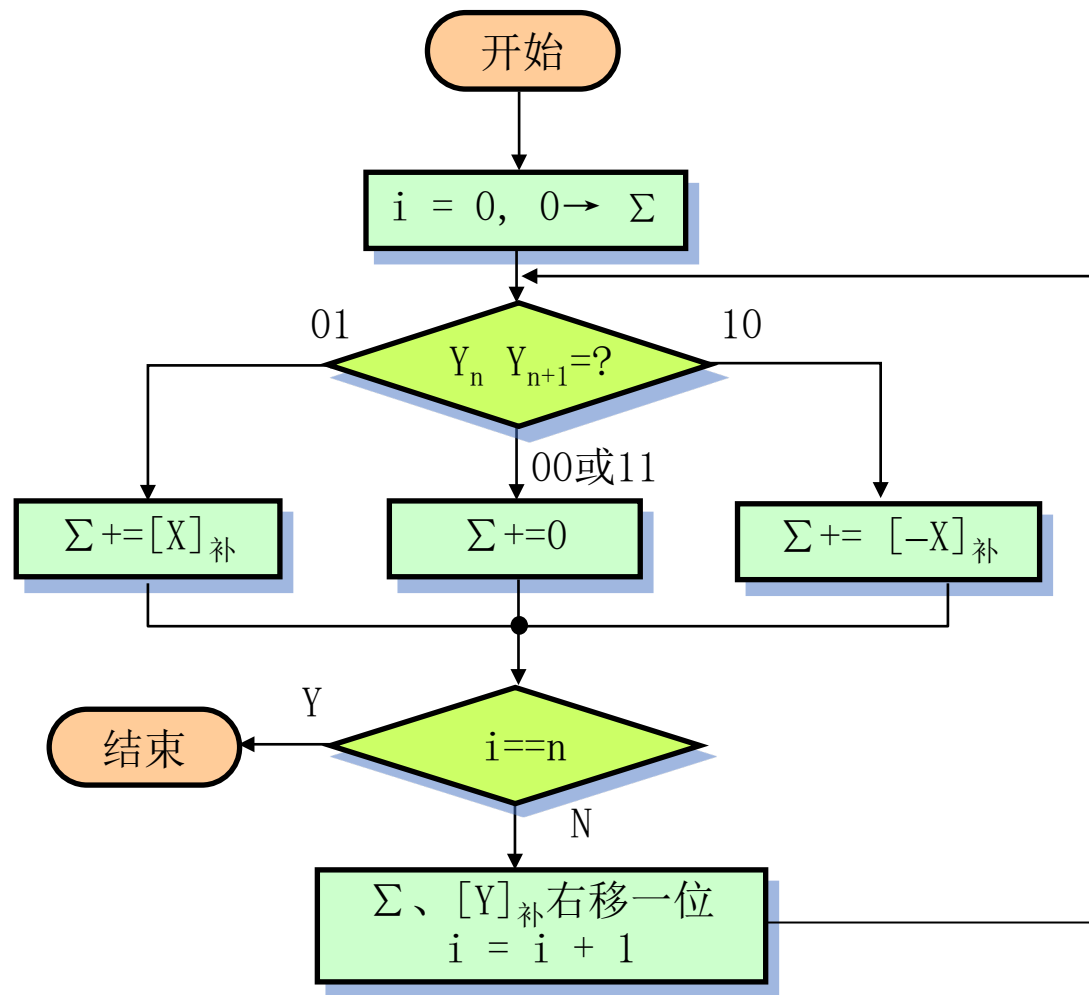


$\{\Sigma, Y\} = \{\Sigma + Y_n |X|, Y\} / 2$ 连同进位位右移

乘法器实验-----原码一位乘法器



乘法器实验-----补码一位乘法器



$$\{\Sigma, Y\} = \{\Sigma + (Y_{n+1} - Y_n)[X]_{\text{补}}, Y\} / 2 \quad \text{算术右移}$$

乘法器实验-----补码一位乘法器

Logisim 2.15.0.2.exe: ☆补码一位乘法器 of alu

File Edit Project Simulate Window Help

alul

- ☆8位可控加减法器
- ☆4位先行进位74182
- ☆4位快速加法器
- ☆16位快速加法器
- ☆32位快速加法器
- ☆5位阵列乘法器
- ☆6位补码阵列乘法器
- ☆5位无符号乘法流水线
- ☆原码一位乘法器
- ☆补码一位乘法器
- ☆算术逻辑运算单元ALU
- Q子电路外观测试
- ★ALU自动测试
- ★5位乘法与门阵列
- ★乘法线流水接口
- ★一位全加器FA封装1
- ★一位全加器FA封装2
- ★一位全加器FA封装3
- ★8位串行加法器
- ★华莱士树

Circuit: ☆补码一位乘法器

Circuit Name ☆补码一位乘法器

Shared Label

Shared Label Facing East

Shared Label Font SansSerif Plain 12

Label Color #000000

Zoom: 120%

输入输出引脚区

功能说明: 补码一位乘法器 乘积=X×Y, 要求采用补码一位乘法实现, 最终结果能输出在乘积引脚上

核心算法: $\{\Sigma, Y\} = \{\Sigma + (Y_{n+1} - Y_n) [X]_{补}, Y\} / 2$

8位补码 Booth一位乘法器

算术右移一位

$\Sigma + (Y_{n+1} - Y_n)_{补}$

ADD

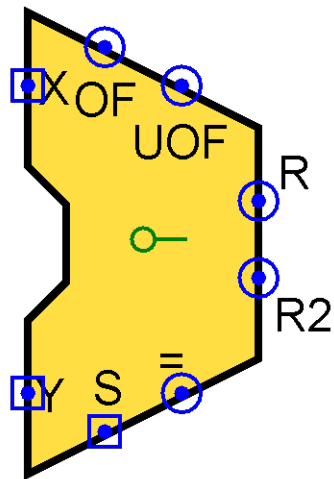
实验三：算术逻辑运算单元ALU实验

算术逻辑运算单元ALU实验

• 实验目的

- 掌握定点数加减法溢出检测方法
 - 熟悉Logisim中各种运算组件的使用方法
 - 逻辑运算部件、乘法器、除法器、移位器
 - 熟悉多路选择器的使用
 - 设计32位ALU
 - 禁止使用Logisim中的加法器、减法器
 - 利用已完成的32位加法器、Logisim其他运算组件构建

算术逻辑运算单元ALU实验



| 引脚 | 输入/输出 | 位宽 | 功能描述 |
|-------|-------|----|--|
| X | 输入 | 32 | 操作数X |
| Y | 输入 | 32 | 操作数Y |
| S | 输入 | 4 | 运算操作码 ALU_OP |
| R | 输出 | 32 | ALU运算结果 |
| R2 | 输出 | 32 | ALU结果第二部分，用于乘法指令结果高位或除法指令的余数位，其它运算时值为零 |
| OF | 输出 | 1 | 有符号加减运算溢出标记，其他运算为0 |
| UOF | 输出 | 1 | 无符号加减运算溢出标记，其它运算时值为零 |
| Equal | 输出 | 1 | Equal=(x==y)?1:0, 对所有运算均有效 |

算术逻辑运算单元ALU实验

| ALU_OP | 十进制 | 运算功能 |
|--------|-----|---|
| 0000 | 0 | Result = X << Y 逻辑左移 (Y取低五位) Result2=0 |
| 0001 | 1 | Result = X >> Y 算术右移 (Y取低五位) Result2=0 |
| 0010 | 2 | Result = X >> Y 逻辑右移 (Y取低五位) Result2=0 |
| 0011 | 3 | Result = (X * Y) _[31:0] Result2 = (X * Y) _[63:32] 无符号乘法 |
| 0100 | 4 | Result = X/Y Result2 = X%Y 无符号除法 |
| 0101 | 5 | Result = X + Y (Set OF/UOF) |
| 0110 | 6 | Result = X - Y (Set OF/UOF) |
| 0111 | 7 | Result = X & Y 按位与 |
| 1000 | 8 | Result = X Y 按位或 |
| 1001 | 9 | Result = X ⊕ Y 按位异或 |
| 1010 | 10 | Result = ~(X Y) 按位或非 |
| 1011 | 11 | Result = (X < Y) ? 1 : 0 有符号比较 |
| 1100 | 12 | Result = (X < Y) ? 1 : 0 无符号比较 |

算术逻辑运算单元ALU实验

Logisim 2.15.0.2.exe: ☆算术逻辑运算单元ALU of alu

File Edit Project Simulate Window Help

alul

- ☆8位可控加减法器
- ☆4位先行进位74182
- ☆4位快速加法器
- ☆16位快速加法器
- ☆32位快速加法器
- ☆5位阵列乘法器
- ☆6位补码阵列乘法器
- ☆5位无符号乘法流水线
- ☆原码一位乘法器
- ☆补码一位乘法器
- ☆算术逻辑运算单元ALU
- Q子电路外观测试
- ★ALU自动测试
- ★5位乘法与门阵列
- ★乘法流水线接口
- ★一位全加器FA封装1
- ★一位全加器FA封装2
- ★一位全加器FA封装3
- ★8位串行加法器
- ★华莱士树

Circuit: ☆算术逻辑运算单元ALU

| | |
|---------------------|--------------------|
| Circuit Name | ☆算术逻辑运算单元ALU |
| Shared Label | ALU |
| Shared Label Facing | North |
| Shared Label Font | SansSerif Plain 12 |
| Label Color | #000000 |

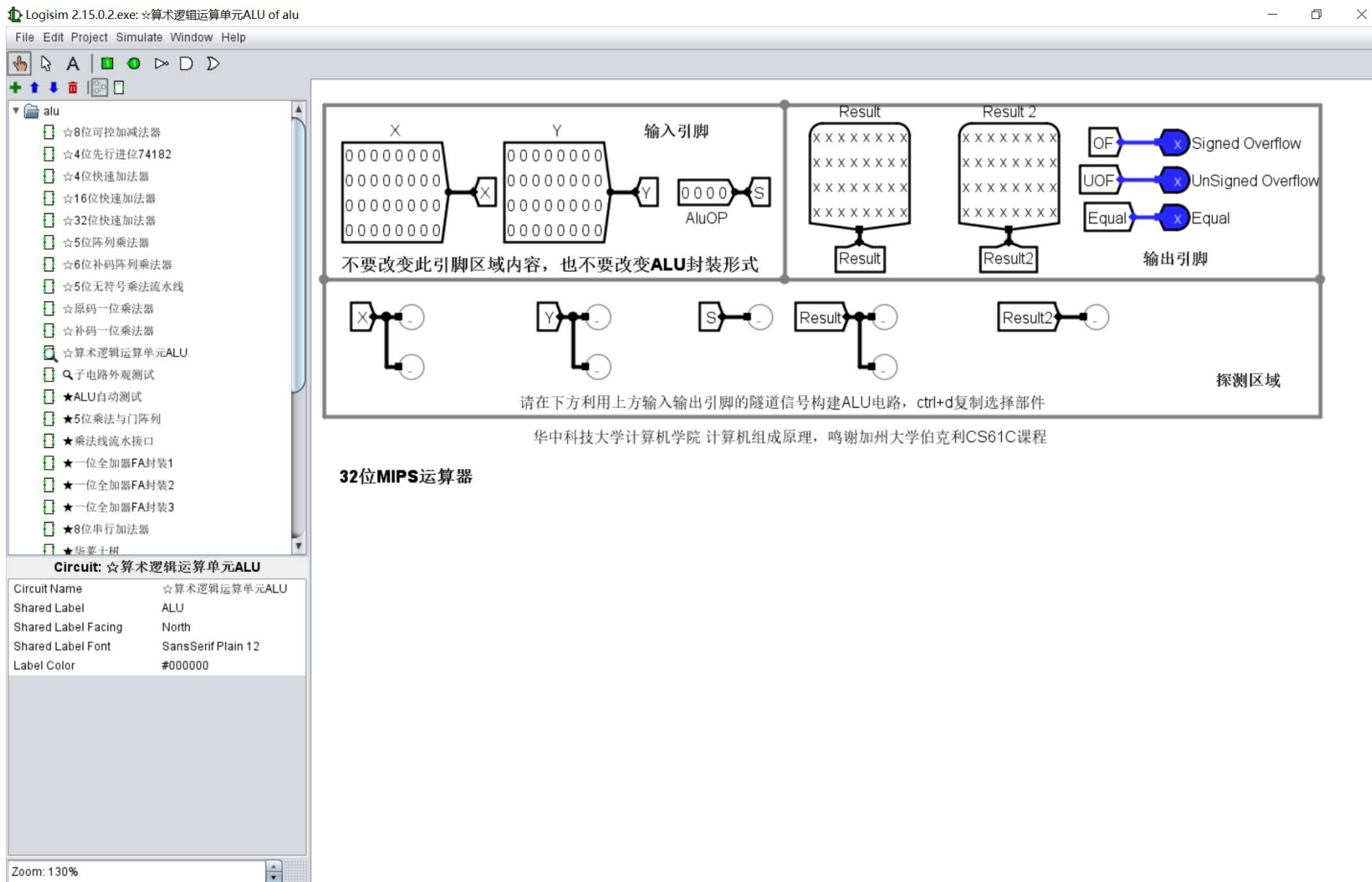
Zoom: 130%

不要改变此引脚区域内容，也不要改变ALU封装形式

请在下方利用上方输入输出引脚的隧道信号构建ALU电路，ctrl+d复制选择部件

华中科技大学计算机学院 计算机组成原理，鸣谢加州大学伯克利CS61C课程

32位MIPS运算器



算术逻辑运算单元ALU实验

Logisim 2.15.0.2.exe: ★ALU自动测试 of alu

File Edit Project Simulate Window Help

alul

- ☆8位可控加减法器
- ☆4位先行进位74182
- ☆4位快速加法器
- ☆16位快速加法器
- ☆32位快速加法器
- ☆5位阵列乘法器
- ☆6位补码阵列乘法器
- ☆5位无符号乘法流水线
- ☆原码一位乘法器
- ☆补码一位乘法器
- ☆算术逻辑运算单元ALU
- Q子电路外观测试
- ★ALU自动测试
- ★5位乘法与门阵列
- ★乘法流水线接口
- ★一位全加器FA封装1
- ★一位全加器FA封装2
- ★一位全加器FA封装3
- ★8位串行加法器
- ★华菱十进制

Circuit: ★ALU自动测试

Circuit Name: ★ALU自动测试

Shared Label: East

Shared Label Facing: SansSerif Plain 12

Label Color: #000000

Zoom: 120%

Trace

操作码为F停止

测试完成

错误记录X

错误记录Y

错误记录ALUOP

成绩

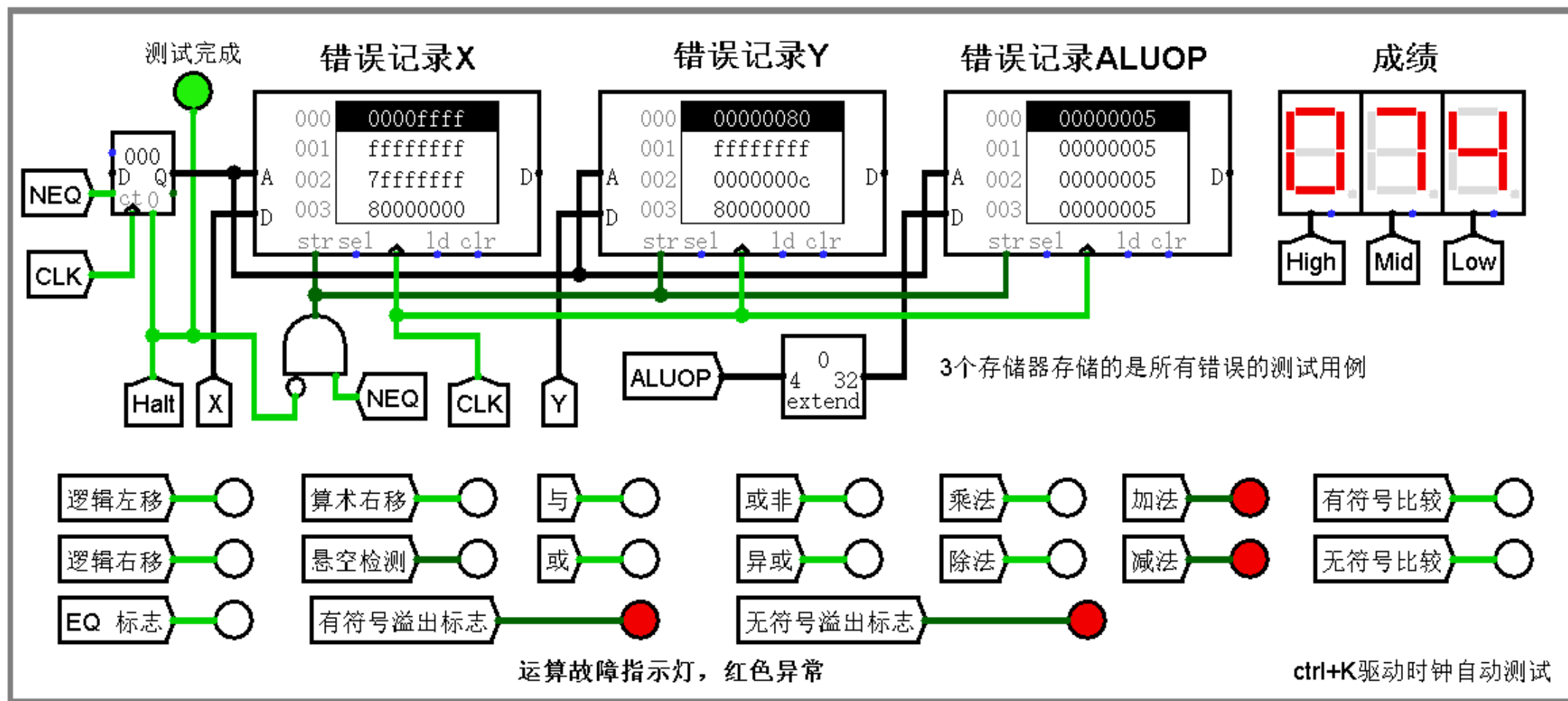
3个存储器存储的是所有错误的测试用例

运算故障指示灯，红色异常

ctrl+k驱动时钟自动测试

华中科技大学：计算机组成课程组

算术逻辑运算单元ALU实验



请同学们开始实验！