Project3 用 circom 实现 poseidon2 哈希算法的电路

一．项目说明

sbox.circom S-box 组件

hush. circom Poseidon2 哈希接口

poseidon2.circom Poseidon2 的核心迭代轮函数

poseidon2 哈希算法参数参考参考文档 1 的 Table1，用(n,t,d)=(256,3,5)或(256,2,5) 2) 电路的公开输入用 poseidon2 哈希值，隐私输入为哈希原象，哈希算法的输入只考虑一个 block 即可。

二.原理结构

sbox.circom

实现了 Poseidon2 中的非线性函数 x↦x5,任何使用 S-box 的地方都可以复用这个组件，比如全轮（full round）或部分轮（partial round）。

原理： 在密码学哈希函数里，为了保证抗碰撞和扩散，每轮除了线性混合（MDS 矩阵）之外，还必须有非线性层。

Poseidon2 的非线性层就是简单的 x^d 幂运算,在 Circom 里通过逐步连乘来实现 x5=x*x*x*x*x

poseidon2.circom

这是 Poseidon2 的核心迭代轮函数，输入一个 **state 向量**，经过多轮非线性 + MDS 线性混合后输出新的 state，不直接处理哈希输出，只是单次 permutation。

过程：

1.轮分配：在全部轮中对所有 state 元素都应用 S-box，部分轮只对 state[0] 应用 S-box，每轮都加上 round constants（RC），然后做 MDS 矩阵混合。

2.线性混合（MDS 矩阵）：Circom 用矩阵乘法把每轮的 state 混合，确保每个元素都依赖于其他元素，实现扩散。

main.circom

构建完整的单 block Poseidon2 哈希接口：公开输入用 poseidon2 哈希值，隐私输入为哈希原象。

调用 poseidon2.circom 里的 permutation，把原像吸收到 sponge state 中，然后生成哈希输出并与公开输入 pubHash 比较。

具体实现：

1.Sponge 初始化：state = [capacity, rate0, rate1]，这里 capacity 放在 state[0]中，rate 放在 state[1..]中。

2.调用 Poseidon2 permutation

把 state 传给 poseidon2.circom，得到新的状态。

3.公共验证

只暴露 state[0] 作为哈希输出，通过 pubHash === out0 约束，让证明者必须提供与 pubHash 匹配的原像。