I am building example with noir and using pedersen for hashing, its simple project (I also inherited some code from battleship)

its comparing in circuit:

fn main(hash: pub Field, mines: [Field; 5]) { let computed_hash = std::hash::pedersen(mines); assert(hash == computed_hash[0]); }

and this is script i am using to generate witness:

(async () => { const barretenberg = await BarretenbergWasm.new(); const pedersen = new SinglePedersen(barretenberg); // Number array of length 15 for sip coordinates (all values must be below 9 with every third either // 0 or 1 to represent orientation const mines = [1, 2, 3, 4, 5]; // Coordinate array must have values coverted to a 32 bytes hex string for Barretenberg Pedersen to match Noir's // implementation. Returns a buffer const mineBuffer = pedersen.compressInputs(mines.map(mine => Buffer.from(numToHex(mine), 'hex'))); // Convert pedersen buffer to hex string and prefix with "0x" to create hash const hash = 0x${mineBuffer.toString('hex')} // Convert to TOML and write witness to prover.toml and public inputs to verified writeFileSync('circuits/board/Prover.toml', stringify({ hash, mines })); console.log('Board witness written to /board/Prover.toml'); writeFileSync('circuits/board/Verifier.toml', stringify({ setpub: [], hash, })); console.log('Board verifier written to /board/Verifier.toml'); })();

but cant satisfy all constrains

[

image

1132×298 25.5 KB

](https://europe1.discourse-cdn.com/business20/uploads/aztec/original/1X/b766f2b32e479bf074e2e55012d095c4ff7558c3.png)