# Debug and test circuits

## Common errors

The most common error you may get while trying to create a proof is:

constraint is not satisfied: [(.. * ..) != (.. * ..) + (.. * ..) + (.. * ..)] The error means the solver couldn't satisfy at least one of the constraints with the provided witness.

note In some cases, you may encounter acouldn't solve computational constraint error, which means the solver couldn't perform an operation needed to verify a constraint. For example, a division by 0. tip You can run the program with-tags=debug to display a more verbose stack trace. Test Engine For a faster development workflow, it is possible (and recommended) toexecute a circuit without running a zk-SNARK prover.

err := test . IsSolved ( circuit , witness , field ) Under the hood it executes the instructions in the circuit, in plain Go, without generating a constraint system and calling a solver.

### Print values

The easiest way to debug a circuit is to useapi.Println() , which behaves likefmt.Println , except it outputs the values when they are solved. For example:

api . Println ( "A.X" , pubKey . A . X ) note With solving errors andapi.Println ,gnark outputs a stack trace which contain the exact line number to refer to in the circuit definition.

## Test

You can implement tests as Go unit tests, in a_test.go file. For example:

// assert object wrapping testing.T assert := test . NewAssert ( t )

// declare the circuit var cubicCircuit Circuit

assert . ProverFailed ( & cubicCircuit ,

& Circuit { PreImage :

42 , Hash :

42 , } )

assert . ProverSucceeded ( & cubicCircuit ,

& Circuit { PreImage :

35 , Hash :

"16130099170765464552823636852555369511329944820189892919423002775646948828469" , } , test . WithCurves ( ecc . BN254 ) ) See thetest package documentation for more details.

In particular, the default behavior of the assert helper is to test the circuit across all supported curves and backends, ensure correct serialization, and cross-test the constraint system solver against abig.Int test execution engine. Edit this page Last updatedonMar 2, 2023 byaybehrouz Previous Compile circuits Next Create and verify proofs * Common errors * * Print values * Test