zkmove

Introduction to zkMove

Guangyu Zhu
2022/6/10

zkmove

What’s zkMove?

- A zero-knowledge proof friendly Move language runtime environment
- We can build scaling and privacy solution based on it

zkmove

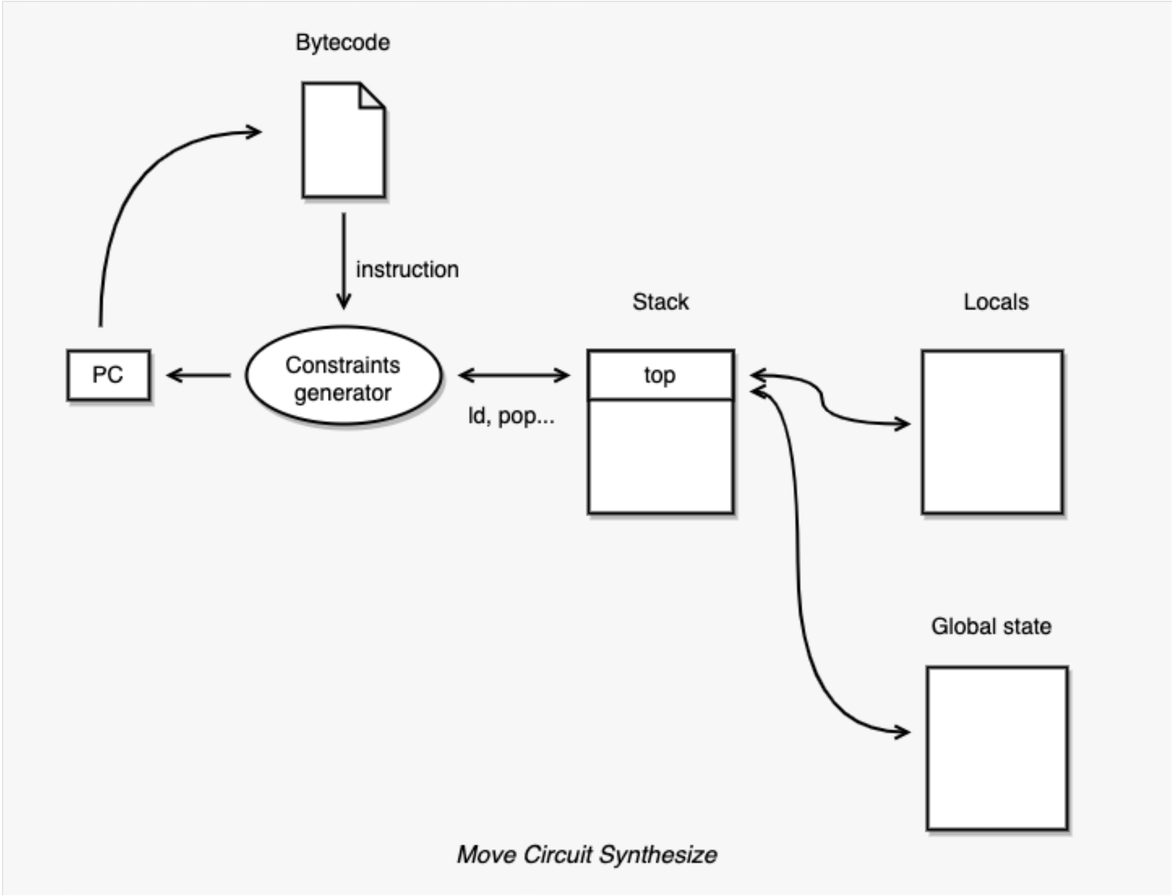
Overview

- A zero-knowledge proof-friendly bytecode virtual machine
  - Compatible with the community Move language virtual machine
  - To solve zero-knowledge proof programmability and composability problems
- Powered by Move language and Halo2
  - Guarantees security of assets at the language level
  - Plonkish arithmetization, Ultra-Plonk, No trusted setup required
- No compromise on performance while pursuing Turing completeness
  - VM circuits to handle conditional branches and loops, can be universal
  - Move circuit directly compiled from bytecode, offer smaller proof size and proving time

zkmove

Move Circuit

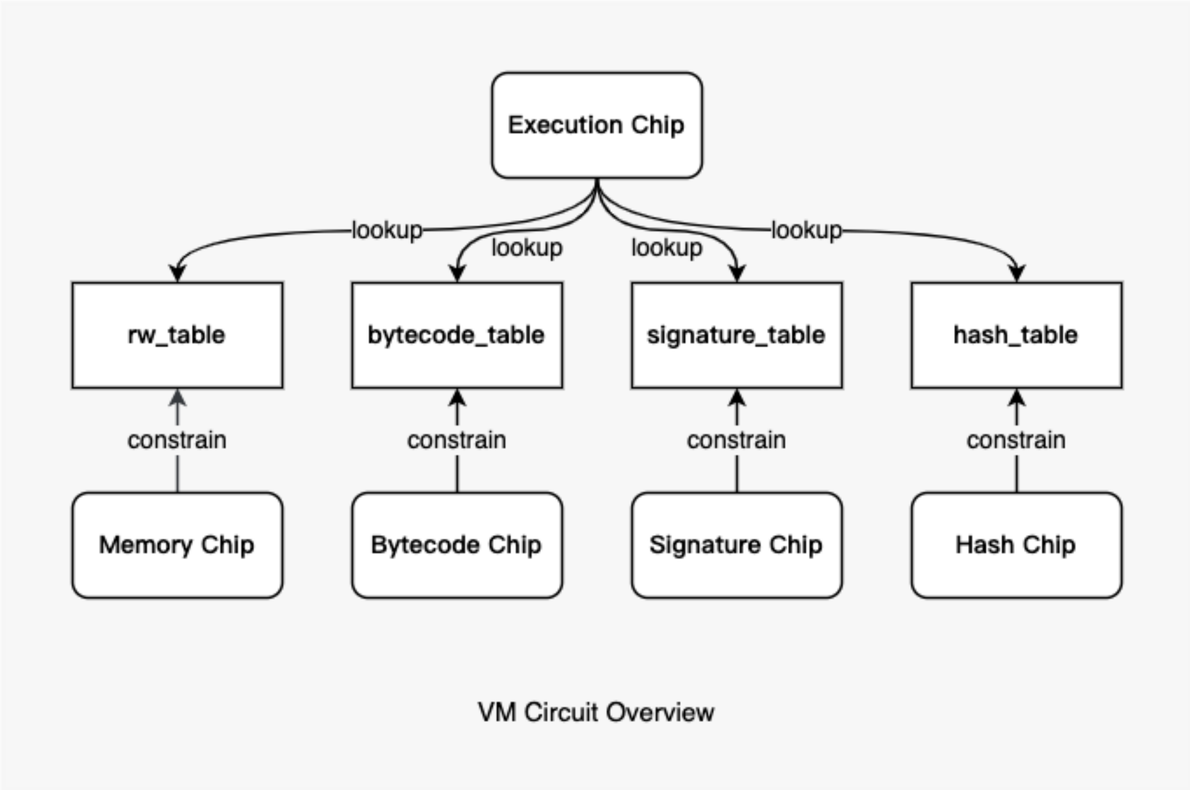
- Application specific circuit
- Constraints are automatically applied by the virtual machine
- Variables and constants in operand stack and locals are Fields value
- ProgramBlock is introduced to handle conditional branch
- Turing-incompleteness, fit for simple transaction such as token transfer



zkmove

VM Circuit

- Inspired by Tinyram、zkEVMs, to verify the consistency and integrity of each step in execution trace
- Lookup bytecode table and use Bytecode Chip to constrain the right byte code was executed
- Lookup read/write table and use Memory Chip to constrain memory coherence (stack, locals, global state)
    - Sorted r/w operations by address



VM Circuit Overview

zkmove

Demo & Performance

- https://github.com/zkmove/zkmove

zkmove

Limitations and issues

- The project is in very early stages
- Lack of support for global state
- Only simple data types are supported
- Underlying proof system needs further improve
- …

Thanks!