# Antshares Virtual Machine

# 1. 简介

# 2. 约定

# 3. 系统组成

# 4. 指令集

## 4.1. 常数指令

### 4.1.1. PUSH0

| | |
|---|---|
| 指令： | PUSH0 |
| 字节码： | 0x00 |
| 别名： | PUSHF 是 PUSH0 的别名 |
| 功能： | 向计算栈中压入一个长度为 0 的字节数组。 |

### 4.1.2. PUSHBYTES

| | |
|---|---|
| 指令： | PUSHBYTES1~PUSHBYTES75 |
| 字节码： | 0x01~0x4b |
| 功能： | 向计算栈中压入一个字节数组，其长度等于本指令字节码的数值。 |

### 4.1.3. PUSHDATA

| | |
|---|---|
| 指令： | PUSHDATA1, PUSHDATA2, PUSHDATA4 |
| 字节码： | 0x4c, 0x4d, 0x4e |
| 功能： | 向计算栈中压入一个字节数组，其长度由本指令后的 1\|2\|4 字节指定。 |

### 4.1.4. PUSHM1

| 指令： | PUSHM1 |
| --- | --- |
| 字节码： | 0x4f |
| 功能： | 向计算栈中压入一个大整数，其数值等于-1。 |

### 4.1.5. PUSHN

| 指令： | PUSH1~PUSH16 |
| --- | --- |
| 字节码： | 0x51~0x60 |
| 别名： | PUSHT 是 PUSH1 的别名 |
| 功能： | 向计算栈中压入一个大整数，其数值等于 1~16。 |

## 4.2. 逻辑控制指令

### 4.2.1. NOP

| 指令： | NOP |
| --- | --- |
| 字节码： | 0x61 |
| 功能： | 没有任何额外的功能，但是会使指令计步器加 1。 |

### 4.2.2. JMP

| 指令： | JMP |
| --- | --- |
| 字节码： | 0x62 |
| 功能： | 无条件跳转到指定偏移位置，偏移量由本指令后的 2 字节指定。 |

### 4.2.3. JMPIF

| 指令： | JMPIF |
| --- | --- |
| 字节码： | 0x63 |
| 功能： | 当计算栈栈顶元素不等于 0 时，跳转到指定偏移位置，偏移量由本指令后的 2 字节指定。不论条件判断成功与否，栈顶元素将被移除。 |

### 4.2.4. JMPIFNOT

| 指令： | JMPIFNOT |
| --- | --- |
| 字节码： | 0x64 |

| | |
|---|---|
| 功能： | 当计算栈栈顶元素等于 0 时，跳转到指定偏移位置，偏移量由本指令后的 2 字节指定。不论条件判断成功与否，栈顶元素将被移除。 |

### 4.2.5. CALL

| | |
|---|---|
| 指令： | CALL |
| 字节码： | 0x65 |
| 功能： | 调用指定偏移位置的函数，偏移量由本指令后的 2 字节指定。 |

### 4.2.6. RET

| | |
|---|---|
| 指令： | RET |
| 字节码： | 0x66 |
| 功能： | 移除调用栈的顶部元素，并使程序在调用栈的下一帧中继续执行。如果调用栈为空，则虚拟机进入停机状态。 |

### 4.2.7. APPCALL

| | |
|---|---|
| 指令： | APPCALL |
| 字节码： | 0x67 |
| 功能： | 调用指定地址的函数，函数地址由本指令后的 20 字节指定。 |

### 4.2.8. SYSCALL

| | |
|---|---|
| 指令： | SYSCALL |
| 字节码： | 0x68 |
| 功能： | 调用指定的互操作函数，函数名称由本指令后的字符串指定。 |

### 4.2.9. TAILCALL

| | |
|---|---|
| 指令： | TAILCALL |
| 字节码： | 0x69 |
| 功能： | 以尾调用的方式，调用指定的互操作函数，函数名称由本指令后的字符串指定。 |

## 4.3. 栈操作指令

### 4.3.1. TOALTSTACK

| | |
|---|---|
| 指令： | TOALTSTACK |
| 字节码： | 0x6b |
| 功能： | 移除计算栈栈顶的元素，并将其压入备用栈。 |

### 4.3.2. FROMALTSTACK

| | |
|---|---|
| 指令： | FROMALTSTACK |
| 字节码： | 0x6c |
| 功能： | 移除备用栈栈顶的元素，并将其压入计算栈。 |

### 4.3.3. XDROP

| | |
|---|---|
| 指令： | XDROP |
| 字节码： | 0x6d |
| 功能： | 移除计算栈栈顶的元素 n，并移除剩余的索引为 n 的元素。 |
| 输入： | $X_n$ $X_{n-1}$ ... $X_2$ $X_1$ $X_0$ n |
| 输出： | $X_{n-1}$ ... $X_2$ $X_1$ $X_0$ |

### 4.3.4. XSWAP

| | |
|---|---|
| 指令： | XSWAP |
| 字节码： | 0x72 |
| 功能： | 移除计算栈栈顶的元素 n，并将剩余的索引为 0 的元素和索引为 n 的元素交换位置。 |
| 输入： | $X_n$ $X_{n-1}$ ... $X_2$ $X_1$ $X_0$ n |
| 输出： | $X_0$ $X_{n-1}$ ... $X_2$ $X_1$ $X_n$ |

### 4.3.5. XTUCK

| | |
|---|---|
| 指令： | XTUCK |
| 字节码： | 0x73 |
| 功能： | 移除计算栈栈顶的元素 n，并将剩余的索引为 0 的元素复制并插入到索引为 n 的位置。 |
| 输入： | $X_n$ $X_{n-1}$ ... $X_2$ $X_1$ $X_0$ n |
| 输出： | $X_n$ $X_0$ $X_{n-1}$ ... $X_2$ $X_1$ $X_0$ |

### 4.3.6. DEPTH

| 指令： | DEPTH |
| --- | --- |
| 字节码： | 0x74 |
| 功能： | 将当前计算栈中的元素数量压入计算栈顶。 |

### 4.3.7. DROP

| 指令： | DROP |
| --- | --- |
| 字节码： | 0x75 |
| 功能： | 移除计算栈栈顶的元素。 |

### 4.3.8. DUP

| 指令： | DUP |
| --- | --- |
| 字节码： | 0x76 |
| 功能： | 复制计算栈栈顶的元素。 |
| 输入： | X |
| 输出： | X X |

### 4.3.9. NIP

| 指令： | NIP |
| --- | --- |
| 字节码： | 0x77 |
| 功能： | 移除计算栈栈顶的第 2 个元素。 |
| 输入： | $X_1$ $X_0$ |
| 输出： | $X_0$ |

### 4.3.10. OVER

| 指令： | OVER |
| --- | --- |
| 字节码： | 0x78 |
| 功能： | 复制计算栈栈顶的第二个元素，并压入栈顶。 |
| 输入： | $X_1$ $X_0$ |
| 输出： | $X_1$ $X_0$ $X_1$ |

### 4.3.11. PICK

| 指令： | PICK |
| --- | --- |

| | |
|---|---|
| 字节码： | 0x79 |
| 功能： | 移除计算栈栈顶的元素 n，并将剩余的索引为 n 的元素复制到栈顶。 |
| 输入： | $X_n$ $X_{n-1}$ … $X_2$ $X_1$ $X_0$ n |
| 输出： | $X_n$ $X_{n-1}$ … $X_2$ $X_1$ $X_0$ $X_n$ |

### 4.3.12. ROLL

| | |
|---|---|
| 指令： | ROLL |
| 字节码： | 0x7a |
| 功能： | 移除计算栈栈顶的元素 n，并将剩余的索引为 n 的元素移动到栈顶。 |
| 输入： | $X_n$ $X_{n-1}$ … $X_2$ $X_1$ $X_0$ n |
| 输出： | $X_{n-1}$ … $X_2$ $X_1$ $X_0$ $X_n$ |

### 4.3.13. ROT

| | |
|---|---|
| 指令： | ROT |
| 字节码： | 0x7b |
| 功能： | 移除计算栈栈顶的第 3 个元素，并将其压入栈顶。 |
| 输入： | $X_2$ $X_1$ $X_0$ |
| 输出： | $X_1$ $X_0$ $X_2$ |

### 4.3.14. SWAP

| | |
|---|---|
| 指令： | SWAP |
| 字节码： | 0x7c |
| 功能： | 交换计算栈栈顶两个元素的位置。 |
| 输入： | $X_1$ $X_0$ |
| 输出： | $X_0$ $X_1$ |

### 4.3.15. TUCK

| | |
|---|---|
| 指令： | TUCK |
| 字节码： | 0x7d |
| 功能： | 复制计算栈栈顶的元素到索引为 2 的位置。 |
| 输入： | $X_1$ $X_0$ |
| 输出： | $X_0$ $X_1$ $X_0$ |

## 4.4. 字符串指令

### 4.4.1. CAT

| | |
|---|---|
| 指令： | CAT |
| 字节码： | 0x7e |
| 功能： | 移除计算栈栈顶的两个元素，并将其拼接后压入栈顶。 |
| 输入： | $X_1$ $X_0$ |
| 输出： | Concat($X_1$, $X_0$) |

### 4.4.2. SUBSTR

| | |
|---|---|
| 指令： | SUBSTR |
| 字节码： | 0x7f |
| 功能： | 移除计算栈栈顶的三个元素，取子串后压入栈顶。 |
| 输入： | X index len |
| 输出： | SubString(X, index, len) |

### 4.4.3. LEFT

| | |
|---|---|
| 指令： | LEFT |
| 字节码： | 0x80 |
| 功能： | 移除计算栈栈顶的两个元素，取子串后压入栈顶。 |
| 输入： | X len |
| 输出： | Left(X, len) |

### 4.4.4. RIGHT

| | |
|---|---|
| 指令： | RIGHT |
| 字节码： | 0x81 |
| 功能： | 移除计算栈栈顶的两个元素，取子串后压入栈顶。 |
| 输入： | X len |
| 输出： | Right(X, len) |

### 4.4.5. SIZE

| | |
|---|---|
| 指令： | SIZE |
| 字节码： | 0x82 |
| 功能： | 将计算栈栈顶元素的长度压入栈顶。 |

| | |
|---|---|
| 输入： | X |
| 输出： | X len(X) |

## 4.5. 逻辑运算指令

### 4.5.1. INVERT

| | |
|---|---|
| 指令： | INVERT |
| 字节码： | 0x83 |
| 功能： | 对计算栈栈顶的元素按位取反。 |
| 输入： | X |
| 输出： | ~X |

### 4.5.2. AND

| | |
|---|---|
| 指令： | AND |
| 字节码： | 0x84 |
| 功能： | 对计算栈栈顶的两个元素执行按位与运算。 |
| 输入： | A B |
| 输出： | A&B |

### 4.5.3. OR

| | |
|---|---|
| 指令： | OR |
| 字节码： | 0x85 |
| 功能： | 对计算栈栈顶的两个元素执行按位或运算。 |
| 输入： | A B |
| 输出： | A\|B |

### 4.5.4. XOR

| | |
|---|---|
| 指令： | XOR |
| 字节码： | 0x86 |
| 功能： | 对计算栈栈顶的两个元素执行按位异或运算。 |
| 输入： | A B |
| 输出： | A^B |

### 4.5.5. EQUAL

| 指令： | EQUAL |
| --- | --- |
| 字节码： | 0x87 |
| 功能： | 对计算栈栈顶的两个元素执行逐字节的相等判断。 |
| 输入： | A B |
| 输出： | Equals(A, B) |

## 4.6. 算数运算指令

### 4.6.1. INC

| 指令： | INC |
| --- | --- |
| 字节码： | 0x8b |
| 功能： | 对计算栈栈顶的大整数执行递增运算。 |
| 输入： | X |
| 输出： | X+1 |

### 4.6.2. DEC

| 指令： | DEC |
| --- | --- |
| 字节码： | 0x8c |
| 功能： | 对计算栈栈顶的大整数执行递减运算。 |
| 输入： | X |
| 输出： | X-1 |

### 4.6.3. SAL

| 指令： | SAL |
| --- | --- |
| 字节码： | 0x8d |
| 功能： | 对计算栈栈顶的大整数执行乘以 2 的运算。 |
| 输入： | X |
| 输出： | X*2 |

### 4.6.4. SAR

| 指令： | SAR |
| --- | --- |
| 字节码： | 0x8e |
| 功能： | 对计算栈栈顶的大整数执行除以 2 的运算。 |

| | |
|---|---|
| 输入： | X |
| 输出： | X/2 |

### 4.6.5. NEGATE

| | |
|---|---|
| 指令： | NEGATE |
| 字节码： | 0x8f |
| 功能： | 求计算栈栈顶的大整数的相反数。 |
| 输入： | X |
| 输出： | -X |

### 4.6.6. ABS

| | |
|---|---|
| 指令： | ABS |
| 字节码： | 0x90 |
| 功能： | 求计算栈栈顶的大整数的绝对值。 |
| 输入： | X |
| 输出： | Abs(X) |

### 4.6.7. NOT

| | |
|---|---|
| 指令： | NOT |
| 字节码： | 0x91 |
| 功能： | 对计算栈栈顶的元素执行逻辑非运算。 |
| 输入： | X |
| 输出： | !X |

### 4.6.8. NZ

| | |
|---|---|
| 指令： | NZ |
| 字节码： | 0x92 |
| 功能： | 判断计算栈栈顶的大整数是否为非 0 值。 |
| 输入： | X |
| 输出： | X!=0 |

### 4.6.9. ADD

| | |
|---|---|
| 指令： | ADD |
| 字节码： | 0x93 |
| 功能： | 对计算栈栈顶的两个大整数执行加法运算。 |

| | |
|---|---|
| 输入： | A B |
| 输出： | A+B |

### 4.6.10. SUB

| | |
|---|---|
| 指令： | SUB |
| 字节码： | 0x94 |
| 功能： | 对计算栈栈顶的两个大整数执行减法运算。 |
| 输入： | A B |
| 输出： | A-B |

### 4.6.11. MUL

| | |
|---|---|
| 指令： | MUL |
| 字节码： | 0x95 |
| 功能： | 对计算栈栈顶的两个大整数执行乘法运算。 |
| 输入： | A B |
| 输出： | A*B |

### 4.6.12. DIV

| | |
|---|---|
| 指令： | DIV |
| 字节码： | 0x96 |
| 功能： | 对计算栈栈顶的两个大整数执行除法运算。 |
| 输入： | A B |
| 输出： | A/B |

### 4.6.13. MOD

| | |
|---|---|
| 指令： | MOD |
| 字节码： | 0x97 |
| 功能： | 对计算栈栈顶的两个大整数执行求余运算。 |
| 输入： | A B |
| 输出： | A%B |

### 4.6.14. SHL

| | |
|---|---|
| 指令： | SHL |
| 字节码： | 0x98 |
| 功能： | 对计算栈中的大整数执行左移运算。 |

| | |
|---|---|
| 输入： | X n |
| 输出： | X<<n |

### 4.6.15. SHR

| | |
|---|---|
| 指令： | SHR |
| 字节码： | 0x99 |
| 功能： | 对计算栈中的大整数执行右移运算。 |
| 输入： | X n |
| 输出： | X>>n |

### 4.6.16. BOOLAND

| | |
|---|---|
| 指令： | BOOLAND |
| 字节码： | 0x9a |
| 功能： | 对计算栈栈顶的两个元素执行逻辑与运算。 |
| 输入： | A B |
| 输出： | A&&B |

### 4.6.17. BOOLOR

| | |
|---|---|
| 指令： | BOOLOR |
| 字节码： | 0x9b |
| 功能： | 对计算栈栈顶的两个元素执行逻辑或运算。 |
| 输入： | A B |
| 输出： | A\|\|B |

### 4.6.18. NUMEQUAL

| | |
|---|---|
| 指令： | NUMEQUAL |
| 字节码： | 0x9c |
| 功能： | 对计算栈栈顶的两个大整数执行相等判断。 |
| 输入： | A B |
| 输出： | A==B |

### 4.6.19. NUMNOTEQUAL

| | |
|---|---|
| 指令： | NUMNOTEQUAL |
| 字节码： | 0x9e |
| 功能： | 对计算栈栈顶的两个大整数执行不相等判断。 |

| | |
|---|---|
| 输入： | A B |
| 输出： | A!=B |

### 4.6.20. LT

| | |
|---|---|
| 指令： | LT |
| 字节码： | 0x9f |
| 功能： | 对计算栈栈顶的两个大整数执行小于判断。 |
| 输入： | A B |
| 输出： | A<B |

### 4.6.21. GT

| | |
|---|---|
| 指令： | GT |
| 字节码： | 0xa0 |
| 功能： | 对计算栈栈顶的两个大整数执行大于判断。 |
| 输入： | A B |
| 输出： | A>B |

### 4.6.22. LTE

| | |
|---|---|
| 指令： | LTE |
| 字节码： | 0xa1 |
| 功能： | 对计算栈栈顶的两个大整数执行小于等于判断。 |
| 输入： | A B |
| 输出： | A<=B |

### 4.6.23. GTE

| | |
|---|---|
| 指令： | GTE |
| 字节码： | 0xa2 |
| 功能： | 对计算栈栈顶的两个大整数执行大于等于判断。 |
| 输入： | A B |
| 输出： | A>=B |

### 4.6.24. MIN

| | |
|---|---|
| 指令： | MIN |
| 字节码： | 0xa3 |
| 功能： | 取出计算栈栈顶的两个大整数中的最小值。 |

| | |
|---|---|
| 输入： | A B |
| 输出： | Min(A, B) |

### 4.6.25. MAX

| | |
|---|---|
| 指令： | MAX |
| 字节码： | 0xa4 |
| 功能： | 取出计算栈栈顶的两个大整数中的最大值。 |
| 输入： | A B |
| 输出： | Max(A, B) |

### 4.6.26. WITHIN

| | |
|---|---|
| 指令： | WITHIN |
| 字节码： | 0xa5 |
| 功能： | 判断计算栈中的大整数是否在指定的数值范围内。 |
| 输入： | X A B |
| 输出： | A<=X&&X<B |

## 4.7. 密码学指令

### 4.7.1. SHA1

| | |
|---|---|
| 指令： | SHA1 |
| 字节码： | 0xa7 |
| 功能： | 对计算栈栈顶的元素执行 SHA1 运算。 |
| 输入： | X |
| 输出： | SHA1(X) |

### 4.7.2. SHA256

| | |
|---|---|
| 指令： | SHA256 |
| 字节码： | 0xa8 |
| 功能： | 对计算栈栈顶的元素执行 SHA256 运算。 |
| 输入： | X |
| 输出： | SHA256(X) |

### 4.7.3. HASH160

| | |
|---|---|
| 指令： | HASH160 |
| 字节码： | 0xa9 |
| 功能： | 对计算栈栈顶的元素执行内置的 160 位散列运算。 |
| 输入： | X |
| 输出： | HASH160(X) |

### 4.7.4. HASH256

| | |
|---|---|
| 指令： | HASH256 |
| 字节码： | 0xaa |
| 功能： | 对计算栈栈顶的元素执行内置的 256 位散列运算。 |
| 输入： | X |
| 输出： | HASH256(X) |

### 4.7.5. CHECKSIG

| | |
|---|---|
| 指令： | CHECKSIG |
| 字节码： | 0xac |
| 功能： | 利用计算栈栈顶元素中的签名和公钥，对当前验证对象执行内置的非对称签名验证操作。 |
| 输入： | S K |
| 输出： | Verify(S, K) |

### 4.7.6. CHECKMULTISIG

| | |
|---|---|
| 指令： | CHECKMULTISIG |
| 字节码： | 0xae |
| 功能： | 利用计算栈栈顶元素中的多个签名和公钥，对当前验证对象执行内置的非对称多重签名验证操作。 |
| 输入： | $S_{m-1}$ … $S_2$ $S_1$ $S_0$ m $K_{n-1}$ … $K_2$ $K_1$ $K_0$ n |
| 输出： | V |
| 备注： | 对于任意的 $S_i \in \{S_0, \dots, S_{m-1}\}$，存在一个 $K_j \in \{K_0, \dots, K_{n-1}\}$ 使得 $\text{Verify}(S_i, K_j) == 1$，则 V=1；否则，V=0。 |

## 4.8. 数据结构指令

### 4.8.1. ARRAYSIZE

| | |
|---|---|
| 指令： | ARRAYSIZE |
| 字节码： | 0xc0 |
| 功能： | 获取计算栈栈顶的数组的元素数量。 |
| 输入： | $[X_0\ X_1\ X_2\ \dots\ X_{n-1}]$ |
| 输出： | n |

### 4.8.2. PACK

| | |
|---|---|
| 指令： | PACK |
| 字节码： | 0xc1 |
| 功能： | 将计算栈栈顶的 n 个元素打包成数组。 |
| 输入： | $X_{n-1}\ \dots\ X_2\ X_1\ X_0\ n$ |
| 输出： | $[X_0\ X_1\ X_2\ \dots\ X_{n-1}]$ |

### 4.8.3. UNPACK

| | |
|---|---|
| 指令： | UNPACK |
| 字节码： | 0xc2 |
| 功能： | 将计算栈栈顶的数组拆包成元素序列。 |
| 输入： | $[X_0\ X_1\ X_2\ \dots\ X_{n-1}]$ |
| 输出： | $X_{n-1}\ \dots\ X_2\ X_1\ X_0\ n$ |

### 4.8.4. PICKITEM

| | |
|---|---|
| 指令： | PICKITEM |
| 字节码： | 0xc3 |
| 功能： | 获取计算栈栈顶的数组中的指定元素。 |
| 输入： | $[X_0\ X_1\ X_2\ \dots\ X_{n-1}]$ i |
| 输出： | $X_i$ |

# 5. 互操作服务