OPCDE

2018 COMAE TECHNOLOGIES

bitcoin opcodes

comae technologies

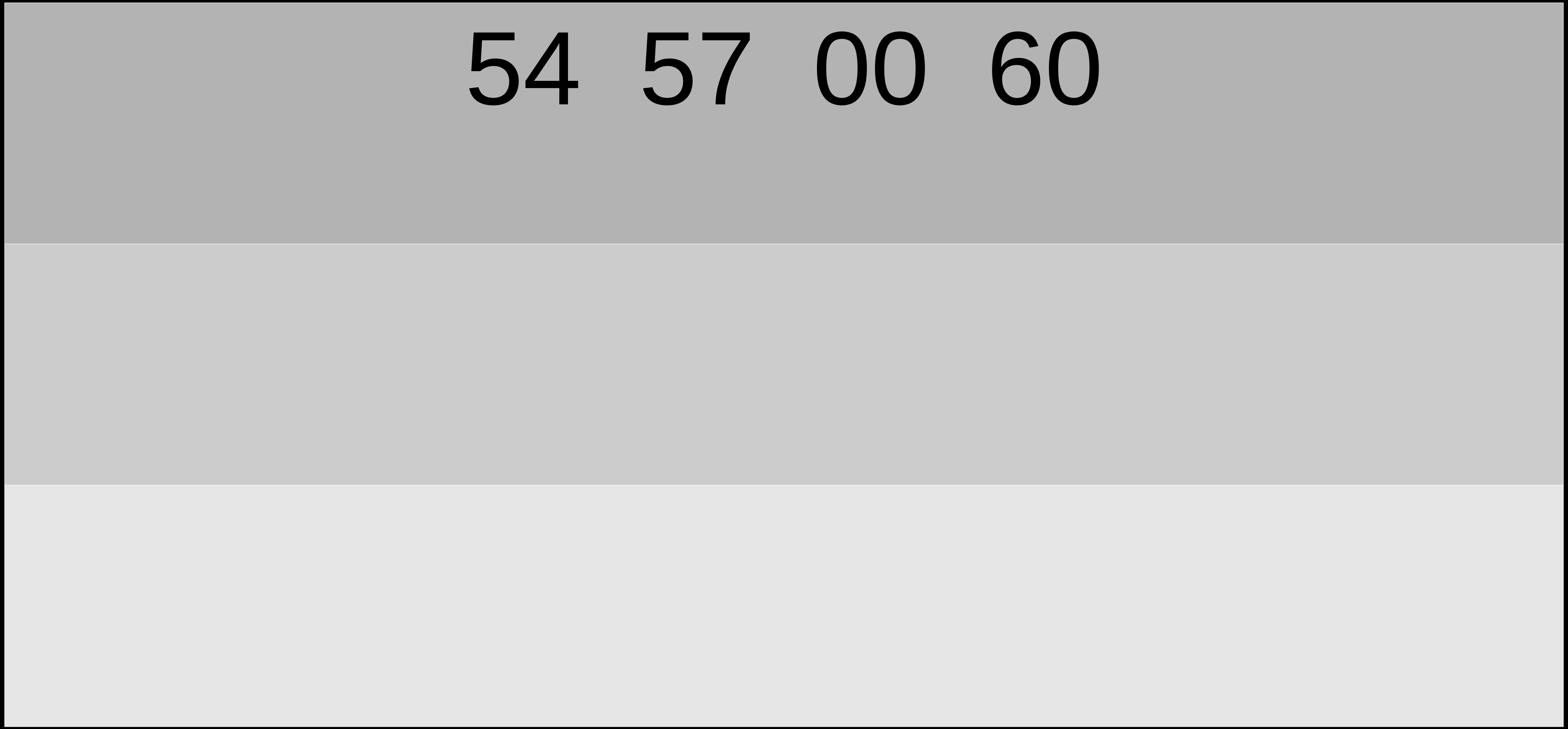
OPCDE.COM

/OPCDE

bitcoin opcodes

opcode	encoding
OP_0	00
OP_1 - OP_16	51 - 60

bitcoin opcodes



54 57 00 60

bitcoin opcodes

54	57	00	60
OP_4	OP_7	OP_0	OP_16

bitcoin opcodes

54 57 00 60

OP_4 OP_7 OP_0 OP_16

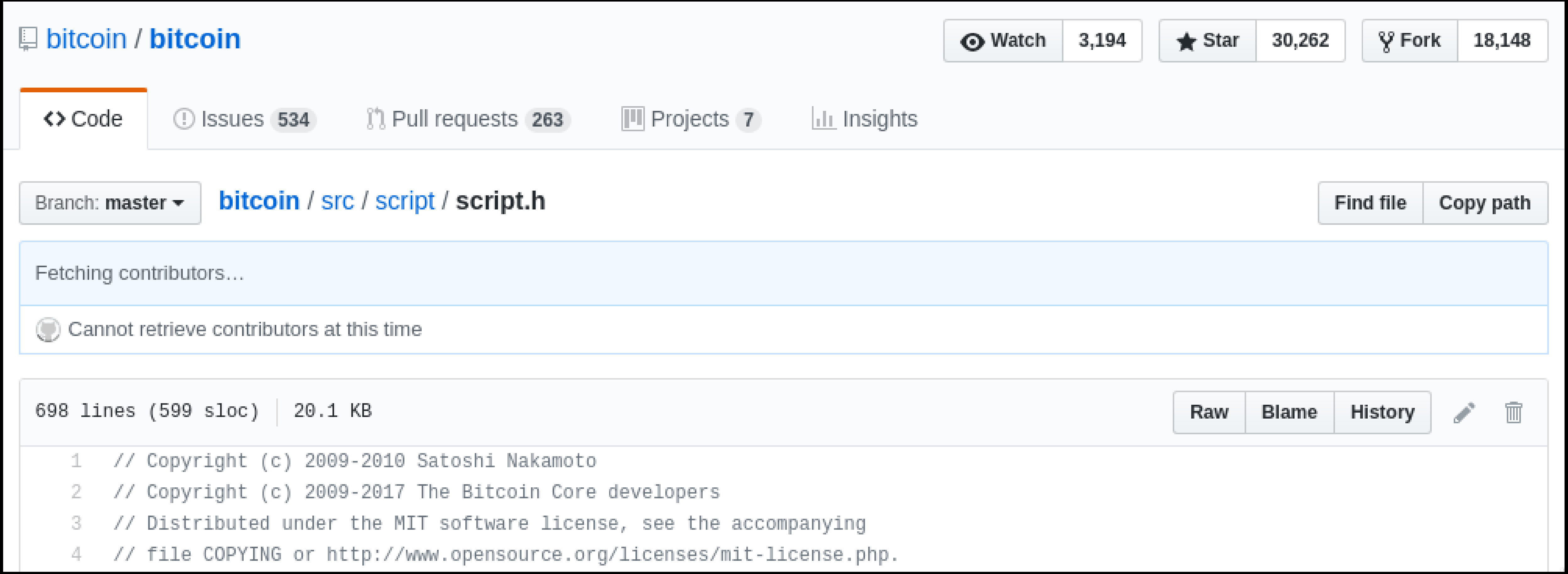
[ ] [4] [4,7] [4,7,0] [4,7,0,16]

bitcoin opcodes

Standard P2PKH (pay-to-public-key-hash)

```
[signature]
[public_key]
OP_DUP
OP_HASH160
[hash160(public_key)]
OP_EQUALVERIFY
OP_CHECKSIG
```

bitcoin opcodes



bitcoin / src / script / script.h

Fetching contributors…

Cannot retrieve contributors at this time

698 lines (599 sloc) | 20.1 KB

```
// Copyright (c) 2009-2010 Satoshi Nakamoto
// Copyright (c) 2009-2017 The Bitcoin Core developers
// Distributed under the MIT software license, see the accompanying
// file COPYING or http://www.opensource.org/licenses/mit-license.php.
```

bitcoin opcodes

```
/** Script opcodes */
enum opcodetype
{

// push value
    OP_0 = 0x00,
    OP_FALSE = OP_0,
    OP_1 = 0x51,
    OP_TRUE=OP_1,
    OP_2 = 0x52,
    OP_3 = 0x53,
    OP_4 = 0x54,
    OP_5 = 0x55,
    OP_6 = 0x56,
    OP_PUSHDATA1 = 0x4c,
    OP_PUSHDATA2 = 0x4d,
    OP_PUSHDATA4 = 0x4e,
    OP_1NEGATE = 0x4f,
    OP_RESERVED = 0x50,
    OP_7 = 0x57,
    OP_8 = 0x58,
    OP_9 = 0x59,
    OP_10 = 0x5a,
    OP_11 = 0x5b,
    OP_12 = 0x5c,
    OP_13 = 0x5d,
    OP_14 = 0x5e,
    OP_15 = 0x5f,
    OP_16 = 0x60,

// control
    OP_NOP = 0x61,
    OP_VER = 0x62,
    OP_IF = 0x63,
    OP_NOTIF = 0x64,
    OP_VERIF = 0x65,
    OP_VERNOTIF = 0x66,
    OP_ELSE = 0x67,
    OP_ENDIF = 0x68,
    OP_VERIFY = 0x69,
    OP_RETURN = 0x6a,

// stack ops
    OP_TOALTSTACK = 0x6b,
    OP_FROMALTSTACK = 0x6c,
    OP_2DROP = 0x6d,
    OP_2DUP = 0x6e,
    OP_3DUP = 0x6f,
    OP_2OVER = 0x70,
    OP_2ROT = 0x71,
    OP_2SWAP = 0x72,
    OP_IFDUP = 0x73,
    OP_DEPTH = 0x74,
    OP_DROP = 0x75,
    OP_DUP = 0x76,
    OP_NIP = 0x77,
    OP_OVER = 0x78,
```

bitcoin opcodes

```
    OP_PICK = 0x79,
    OP_ROLL = 0x7a,
    OP_ROT = 0x7b,
    OP_SWAP = 0x7c,
    OP_TUCK = 0x7d,

// splice ops
    OP_CAT = 0x7e,
    OP_SUBSTR = 0x7f,
    OP_LEFT = 0x80,
    OP_RIGHT = 0x81,
    OP_SIZE = 0x82,

// bit logic
    OP_INVERT = 0x83,
    OP_AND = 0x84,
    OP_OR = 0x85,
    OP_XOR = 0x86,
    OP_EQUAL = 0x87,
    OP_EQUALVERIFY = 0x88,
    OP_RESERVED1 = 0x89,
    OP_RESERVED2 = 0x8a,

// numeric
    OP_1ADD = 0x8b,
    OP_1SUB = 0x8c,
    OP_2MUL = 0x8d,
    OP_2DIV = 0x8e,
    OP_NEGATE = 0x8f,
    OP_ABS = 0x90,
    OP_NOT = 0x91,
    OP_0NOTEQUAL = 0x92,
    OP_ADD = 0x93,
    OP_SUB = 0x94,
    OP_MUL = 0x95,
    OP_DIV = 0x96,
    OP_MOD = 0x97,
    OP_LSHIFT = 0x98,
    OP_RSHIFT = 0x99,
    OP_BOOLAND = 0x9a,
    OP_BOOLOR = 0x9b,
    OP_NUMEQUAL = 0x9c,
    OP_NUMEQUALVERIFY = 0x9d,
    OP_NUMNOTEQUAL = 0x9e,
    OP_LESSTHAN = 0x9f,
    OP_GREATERTHAN = 0xa0,
    OP_LESSTHANOREQUAL = 0xa1,
    OP_GREATERTHANOREQUAL = 0xa2,
    OP_MIN = 0xa3,
    OP_MAX = 0xa4,
    OP_WITHIN = 0xa5,
```

bitcoin opcodes

```
// crypto
    OP_RIPEMD160 = 0xa6,
    OP_SHA1 = 0xa7,
    OP_SHA256 = 0xa8,
    OP_HASH160 = 0xa9,
    OP_HASH256 = 0xaa,
    OP_CODESEPARATOR = 0xab,
    OP_CHECKSIG = 0xac,
    OP_CHECKSIGVERIFY = 0xad,
    OP_CHECKMULTISIG = 0xae,
    OP_CHECKMULTISIGVERIFY = 0xaf,

// expansion
    OP_NOP1 = 0xb0,
    OP_CHECKLOCKTIMEVERIFY = 0xb1,
    OP_NOP2 =  OP_CHECKLOCKTIMEVERIFY,
    OP_CHECKSEQUENCEVERIFY = 0xb2,
    OP_NOP3 =  OP_CHECKSEQUENCEVERIFY,
    OP_NOP4 = 0xb3,
    OP_NOP5 = 0xb4,
    OP_NOP6 = 0xb5,
    OP_NOP7 = 0xb6,
    OP_NOP8 = 0xb7,
    OP_NOP9 = 0xb8,
    OP_NOP10 = 0xb9,

    // template matching params
    OP_SMALLINTEGER = 0xfa,
    OP_PUBKEYS = 0xfb,
    OP_PUBKEYHASH = 0xfd,
    OP_PUBKEY = 0xfe,
    OP_INVALIDOPCODE = 0xff,
};
```

bitcoin opcodes

Turing Complete?

bitcoin opcodes

At REcon 2015 Christopher Domas demonstrated M/o/Vfuscator,
a “Turing Complete” compiler using a single opcode.