crytic / evm-opcodes Public

Ethereum opcodes and instruction reference

Apache-2.0 License

608 stars 93 forks

Star Watch

Code Issues 5 Pull requests Actions Projects Wiki Security Insights

master

computerality ... on 14 Sep 2021

View code

Ethereum VM (EVM) Opcodes and Instruction Reference

This reference consolidates EVM opcode information from the yellow paper, stack exchange, solidity source, parity source, evm-opcode-gas-costs and Manticore.

New issues and contributions are welcome, and are covered by bounties from Trail of Bits. Join us in #ethereum on the Empire Hacking Slack to discuss Ethereum security tool development.

Notes

The size of a "word" in EVM is 256 bits.

The gas information is a work in progress. If an asterisk is in the Gas column, the base cost is shown but may vary based on the opcode arguments.

Table

Opcode	Name	Description	Extra Info	Gas
`0x00`	STOP	Halts execution	-	0
`0x01`	ADD	Addition operation	-	3
`0x02`	MUL	Multiplication operation	-	5
`0x03`	SUB	Subtraction operation	-	3
`0x04`	DIV	Integer division operation	-	5
`0x05`	SDIV	Signed integer division operation (truncated)	-	5
`0x06`	MOD	Modulo remainder operation	-	5
`0x07`	SMOD	Signed modulo remainder operation	-	5
`0x08`	ADDMOD	Modulo addition operation	-	8
`0x09`	MULMOD	Modulo multiplication operation	-	8
`0x0a`	EXP	Exponential operation	-	10*
`0x0b`	SIGNEXTEND	Extend length of two's complement signed integer	-	5
`0x0c - 0x0f`	Unused	Unused	-	
`0x10`	LT	Less-than comparison	-	3
`0x11`	GT	Greater-than comparison	-	3
`0x12`	SLT	Signed less-than comparison	-	3
`0x13`	SGT	Signed greater-than comparison	-	3
`0x14`	EQ	Equality comparison	-	3

Opcode	Name	Description	Extra Info	Gas
`0x15`	ISZERO	Simple not operator	-	3
`0x16`	AND	Bitwise AND operation	-	3
`0x17`	OR	Bitwise OR operation	-	3
`0x18`	XOR	Bitwise XOR operation	-	3
`0x19`	NOT	Bitwise NOT operation	-	3
`0x1a`	BYTE	Retrieve single byte from word	-	3
`0x1b`	SHL	Shift Left	EIP145	3
`0x1c`	SHR	Logical Shift Right	EIP145	3
`0x1d`	SAR	Arithmetic Shift Right	EIP145	3
`0x20`	KECCAK256	Compute Keccak-256 hash	-	30*
`0x21 - 0x2f`	Unused	Unused		
`0x30`	ADDRESS	Get address of currently executing account	-	2
`0x31`	BALANCE	Get balance of the given account	-	700
`0x32`	ORIGIN	Get execution origination address	-	2
`0x33`	CALLER	Get caller address	-	2
`0x34`	CALLVALUE	Get deposited value by the instruction/transaction responsible for this execution	-	2
`0x35`	CALLDATALOAD	Get input data of current environment	-	3
`0x36`	CALLDATASIZE	Get size of input data in current environment	-	2*

Opcode	Name	Description	Extra Info	Gas
`0x37`	CALLDATACOPY	Copy input data in current environment to memory	-	3
`0x38`	CODESIZE	Get size of code running in current environment	-	2
`0x39`	CODECOPY	Copy code running in current environment to memory	-	3*
`0x3a`	GASPRICE	Get price of gas in current environment	-	2
`0x3b`	EXTCODESIZE	Get size of an account's code	-	700
`0x3c`	EXTCODECOPY	Copy an account's code to memory	-	700*
`0x3d`	RETURNDATASIZE	Pushes the size of the return data buffer onto the stack	EIP 211	2
`0x3e`	RETURNDATACOPY	Copies data from the return data buffer to memory	EIP 211	3
`0x3f`	EXTCODEHASH	Returns the keccak256 hash of a contract's code	EIP 1052	700
`0x40`	BLOCKHASH	Get the hash of one of the 256 most recent complete blocks	-	20
`0x41`	COINBASE	Get the block's beneficiary address	-	2
`0x42`	TIMESTAMP	Get the block's timestamp	-	2
`0x43`	NUMBER	Get the block's number	-	2

Opcode	Name	Description	Extra Info	Gas
`0x44`	DIFFICULTY	Get the block's difficulty	-	2
`0x45`	GASLIMIT	Get the block's gas limit	-	2
`0x46`	CHAINID	Returns the current chain's EIP-155 unique identifier	EIP 1344	2
`0x47 - 0x4f`	Unused	-		

`0x48`	BASEFEE	current block it is executing in.	3198	2
`0x50`	POP	Remove word from stack	-	2
`0x51`	MLOAD	Load word from memory	-	3*
`0x52`	MSTORE	Save word to memory	-	3*
`0x53`	MSTORE8	Save byte to memory	-	3
`0x54`	SLOAD	Load word from storage	-	800
`0x55`	SSTORE	Save word to storage	-	20000**
`0x56`	JUMP	Alter the program counter	-	8
`0x57`	JUMPI	Conditionally alter the program counter	-	10
`0x58`	GETPC	Get the value of the program counter prior to the increment	-	2
`0x59`	MSIZE	Get the size of active memory in bytes	-	2

Opcode	Name	Description	Extra Info	Gas
`0x5a`	GAS	Get the amount of available gas, including the corresponding reduction for the cost of this instruction	-	2
`0x5b`	JUMPDEST	Mark a valid destination for jumps	-	1
`0x5c - 0x5f`	Unused	-		
`0x60`	PUSH1	Place 1 byte item on stack	-	3
`0x61`	PUSH2	Place 2-byte item on stack	-	3
`0x62`	PUSH3	Place 3-byte item on stack	-	3
`0x63`	PUSH4	Place 4-byte item on stack	-	3
`0x64`	PUSH5	Place 5-byte item on stack	-	3
`0x65`	PUSH6	Place 6-byte item on stack	-	3
`0x66`	PUSH7	Place 7-byte item on stack	-	3
`0x67`	PUSH8	Place 8-byte item on stack	-	3
`0x68`	PUSH9	Place 9-byte item on stack	-	3
`0x69`	PUSH10	Place 10-byte item on stack	-	3
`0x6a`	PUSH11	Place 11-byte item on stack	-	3

Opcode	Name	Description	Extra Info	Gas
`0x6b`	PUSH12	Place 12-byte item on stack	-	3
`0x6c`	PUSH13	Place 13-byte item on stack	-	3
`0x6d`	PUSH14	Place 14-byte item on stack	-	3
`0x6e`	PUSH15	Place 15-byte item on stack	-	3
`0x6f`	PUSH16	Place 16-byte item on stack	-	3
`0x70`	PUSH17	Place 17-byte item on stack	-	3
`0x71`	PUSH18	Place 18-byte item on stack	-	3
`0x72`	PUSH19	Place 19-byte item on stack	-	3
`0x73`	PUSH20	Place 20-byte item on stack	-	3
`0x74`	PUSH21	Place 21-byte item on stack	-	3
`0x75`	PUSH22	Place 22-byte item on stack	-	3
`0x76`	PUSH23	Place 23-byte item on stack	-	3
`0x77`	PUSH24	Place 24-byte item on stack	-	3
`0x78`	PUSH25	Place 25-byte item on stack	-	3
`0x79`	PUSH26	Place 26-byte item on stack	-	3
`0x7a`	PUSH27	Place 27-byte item on stack	-	3

Opcode	Name	Description	Extra Info	Gas
`0x7b`	PUSH28	Place 28-byte item on stack	-	3
`0x7c`	PUSH29	Place 29-byte item on stack	-	3
`0x7d`	PUSH30	Place 30-byte item on stack	-	3
`0x7e`	PUSH31	Place 31-byte item on stack	-	3
`0x7f`	PUSH32	Place 32-byte (full word) item on stack	-	3
`0x80`	DUP1	Duplicate 1st stack item	-	3
`0x81`	DUP2	Duplicate 2nd stack item	-	3
`0x82`	DUP3	Duplicate 3rd stack item	-	3
`0x83`	DUP4	Duplicate 4th stack item	-	3
`0x84`	DUP5	Duplicate 5th stack item	-	3
`0x85`	DUP6	Duplicate 6th stack item	-	3
`0x86`	DUP7	Duplicate 7th stack item	-	3
`0x87`	DUP8	Duplicate 8th stack item	-	3
`0x88`	DUP9	Duplicate 9th stack item	-	3
`0x89`	DUP10	Duplicate 10th stack item	-	3
`0x8a`	DUP11	Duplicate 11th stack item	-	3

Opcode	Name	Description	Extra Info	Gas
`0x8b`	DUP12	Duplicate 12th stack item	-	3
`0x8c`	DUP13	Duplicate 13th stack item	-	3
`0x8d`	DUP14	Duplicate 14th stack item	-	3
`0x8e`	DUP15	Duplicate 15th stack item	-	3
`0x8f`	DUP16	Duplicate 16th stack item	-	3
`0x90`	SWAP1	Exchange 1st and 2nd stack items	-	3
`0x91`	SWAP2	Exchange 1st and 3rd stack items	-	3
`0x92`	SWAP3	Exchange 1st and 4th stack items	-	3
`0x93`	SWAP4	Exchange 1st and 5th stack items	-	3
`0x94`	SWAP5	Exchange 1st and 6th stack items	-	3
`0x95`	SWAP6	Exchange 1st and 7th stack items	-	3
`0x96`	SWAP7	Exchange 1st and 8th stack items	-	3
`0x97`	SWAP8	Exchange 1st and 9th stack items	-	3
`0x98`	SWAP9	Exchange 1st and 10th stack items	-	3
`0x99`	SWAP10	Exchange 1st and 11th stack items	-	3
`0x9a`	SWAP11	Exchange 1st and 12th stack items	-	3

Opcode	Name	Description	Extra Info	Gas
`0x9b`	SWAP12	Exchange 1st and 13th stack items	-	3
`0x9c`	SWAP13	Exchange 1st and 14th stack items	-	3
`0x9d`	SWAP14	Exchange 1st and 15th stack items	-	3
`0x9e`	SWAP15	Exchange 1st and 16th stack items	-	3
`0x9f`	SWAP16	Exchange 1st and 17th stack items	-	3
`0xa0`	LOG0	Append log record with no topics	-	375
`0xa1`	LOG1	Append log record with one topic	-	750
`0xa2`	LOG2	Append log record with two topics	-	1125
`0xa3`	LOG3	Append log record with three topics	-	1500
`0xa4`	LOG4	Append log record with four topics	-	1875
`0xa5` - `0xaf`	Unused	-		
`0xb0`	JUMPTO	Tentative libevmasm has different numbers	EIP 615	
`0xb1`	JUMPIF	Tentative	EIP 615	
`0xb2`	JUMPSUB	Tentative	EIP 615	
`0xb4`	JUMPSUBV	Tentative	EIP 615	
`0xb5`	BEGINSUB	Tentative	EIP 615	

Opcode	Name	Description	Extra Info	Gas
`0xb6`	BEGINDATA	Tentative	EIP 615	
`0xb8`	RETURNSUB	Tentative	EIP 615	
`0xb9`	PUTLOCAL	Tentative	EIP 615	
`0xba`	GETLOCAL	Tentative	EIP 615	
`0xbb - 0xe0`	Unused	-		
`0xe1`	SLOADBYTES	Only referenced in pyethereum	-	-
`0xe2`	SSTOREBYTES	Only referenced in pyethereum	-	-
`0xe3`	SSIZE	Only referenced in pyethereum	-	-
`0xe4 - 0xef`	Unused	-		
`0xf0`	CREATE	Create a new account with associated code	-	32000
`0xf1`	CALL	Message-call into an account	-	Complicated
`0xf2`	CALLCODE	Message-call into this account with alternative account's code	-	Complicated
`0xf3`	RETURN	Halt execution returning output data	-	0

Opcode	Name	Description	Extra Info	Gas
`0xf4`	DELEGATECALL	Message-call into this account with an alternative account's code, but persisting into this account with an alternative account's code	-	Complicated
`0xf5`	CREATE2	Create a new account and set creation address to `sha3(sender + sha3(init code)) % 2**160`	-	
`0xf6 - 0xf9`	Unused	-	-	
`0xfa`	STATICCALL	Similar to CALL, but does not modify state	-	40
`0xfb`	Unused	-	-	
`0xfc`	TXEXECGAS	Not in yellow paper FIXME	-	-
`0xfd`	REVERT	Stop execution and revert state changes, without consuming all provided gas and providing a reason	-	0
`0xfe`	INVALID	Designated invalid instruction	-	0
`0xff`	SELFDESTRUCT	Halt execution and register account for later deletion	-	5000*

Instruction Details

ADD

Takes two words from stack, adds them, then pushes the result onto the stack.

Pseudocode: `push(s[0]+s[1])`

PUSHX

The following X bytes are read from PC, placed into a word, then this word is pushed onto the stack.

CALL

Releases

No releases published

Packages

No packages published

Contributors 9