Value	Mnemonic	Gas Used	Subset	Removed from stack	Added to stack	Notes	Formula	Formula Notes
0x00	STOP	0	zero	0	0	Halts execution.		
0x01	ADD	3	verylow	2	1	Addition operation		
0x02	MUL	5	low	2	1	Multiplication operation.		
0x03	SUB	3	verylow	2	1	Subtraction operation.		
0x04	DIV	5	low	2	1	Integer division operation.		
0x05	SDIV	5	low	2	1	Signed integer division operation (truncated).		
0x06	MOD	5	low	2	1	Modulo remainder operation		
0x07	SMOD	5	low	2	1	Signed modulo remainder operation.		
0x08	ADDMOD	8	mid	3	1	Modulo addition operation.		
0x09	MULMOD	8	mid	3	1	Modulo multiplication operation.		
0x0a	EXP	FORMULA		2	1	Exponential operation.	(exp == 0) ? 10 :	If exponent is 0, gas used is 10. If exponent is greater than 0, gas used is 10 plus 10 times a factor related to how large the log of the exponent is.
0x0b	SIGNEXTEND	5	low	2	1	Extend length of two’s complement signed integer.		
0x10	LT	3	verylow	2	1	Less-than comparison.		
0x11	GT	3	verylow	2	1	Greater-than comparison.		
0x12	SLT	3	verylow	2	1	Signed less-than comparison.		
0x13	SGT	3	verylow	2	1	Signed greater-than comparison.		
0x14	EQ	3	verylow	2	1	Equality comparison.		
0x15	ISZERO	3	verylow	1	1	Simple not operator.		
0x16	AND	3	verylow	2	1	Bitwise AND operation.		
0x17	OR	3	verylow	2	1	Bitwise OR operation		
0x18	XOR	3	verylow	2	1	Bitwise XOR operation.		
0x19	NOT	3	verylow	1	1	Bitwise NOT operation.		
0x1a	BYTE	3	verylow	2	1	Retrieve single byte from word		
0x20	SHA3	FORMULA		2	1	Compute Keccak-256 hash.	30 + 6 * (size of i	30 is the paid for the operation plus 6 paid for each word (rounded up) for the input data.
0x30	ADDRESS	2	base	0	1	Get address of currently executing account.		
0x31	BALANCE	400		1	1	Get balance of the given account.		
0x32	ORIGIN	2	base	0	1	Get execution origination address.		
0x33	CALLER	2	base	0	1	Get caller address.		
0x34	CALLVALUE	2	base	0	1	Get deposited value by the instruction/transaction responsible for this execution.		
0x35	CALLDATALOAD	3	verylow	1	1	Get input data of current environment.		
0x36	CALLDATASIZE	2	base	0	1	Get size of input data in current environment.		
0x37	CALLDATACOPY	FORMULA		3	0	Copy input data in current environment to me	2 + 3 * (number c	2 is paid for the operation plus 3 for each word copied (rounded up).
0x38	CODESIZE	2	base	0	1	Get size of code running in current environment.		
0x39	CODECOPY	FORMULA		3	0	Copy code running in current environment to	2 + 3 * (number c	2 is paid for the operation plus 3 for each word copied (rounded up).
0x3a	GASPRICE	2	base	0	1	Get price of gas in current environment.		
0x3b	EXTCODESIZE	700	extcode	1	1	Get size of an account’s code.		
0x3c	EXTCODECOPY	FORMULA		4	0	Copy an account’s code to memory.	700 + 3 * (numbe	700 is paid for the operation plus 3 for each word copied (rounded up).
0x40	BLOCKHASH	20		1	1	Get the hash of one of the 256 most recent complete blocks.		
0x41	COINBASE	2	base	0	1	Get the block’s beneficiary address.		
0x42	TIMESTAMP	2	base	0	1	Get the block’s timestamp.		
0x43	NUMBER	2	base	0	1	Get the block’s number.		
0x44	DIFFICULTY	2	base	0	1	Get the block’s difficulty.		
0x45	GASLIMIT	2	base	0	1	Get the block’s gas limit.		
0x50	POP	2	base	1	0	Remove item from stack.		
0x51	MLOAD	3	verylow	1	1	Load word from memory.		
0x52	MSTORE	3	verylow	2	0	Save word to memory		
0x53	MSTORE8	3	verylow	2	0	Save byte to memory.		
0x54	SLOAD	200		1	1	Load word from storage		
0x55	SSTORE	FORMULA		1	1	Save word to storage.	((value != 0) && (	20000 is paid when storage value is set to non-zero from zero. 5000 is paid when the storage value's zeroness remains unchanged or is set to zero.
0x56	JUMP	8	mid	1	0	Alter the program counter		
0x57	JUMPI	10	high	2	0	Conditionally alter the program counter.		
0x58	PC	2	base	0	1	Get the value of the program counter prior to the increment corresponding to this instruction.		
0x59	MSIZE	2	base	0	1	Get the size of active memory in bytes.		
0x5a	GAS	2	base	0	1	Get the amount of available gas, including the corresponding reduction for the cost of this instruction.		
0x5b	JUMPDEST	1		0	0	Mark a valid destination for jumps		
0x60 -- 0x7f	PUSH*	3	verylow	0	1	Place * byte item on stack. 0 < * <= 32		
0x80 -- 0x8f	DUP*	3	verylow	*	* + 1	Duplicate *th stack item. 0 < * <= 16		
0x90 -- 0x9f	SWAP*	3	verylow	* + 1	* + 1	Exchange 1st and (* + 1)th stack items.		
0xa0	LOG0	FORMULA		2	0	Append log record with no topics.	375 + 8 * (numbe	375 is paid for operation plus 8 for each byte in data to be logged.
0xa1	LOG1	FORMULA		3	0	Append log record with one topic.	375 + 8 * (numbe	375 is paid for operation plus 8 for each byte in data to be logged plus 375 for the 1 topic to be logged.
0xa2	LOG2	FORMULA		4	0	Append log record with two topics.	375 + 8 * (numbe	375 is paid for operation plus 8 for each byte in data to be logged plus 2 * 375 for the 2 topics to be logged.
0xa3	LOG3	FORMULA		5	0	Append log record with three topics.	375 + 8 * (numbe	375 is paid for operation plus 8 for each byte in data to be logged plus 3 * 375 for the 3 topics to be logged.
0xa4	LOG4	FORMULA		6	0	Append log record with four topics.	375 + 8 * (numbe	375 is paid for operation plus 8 for each byte in data to be logged plus 4 * 375 for the 4 topics to be logged.
0xf0	CREATE	32000		3	1	Create a new account with associated code.		
0xf1	CALL	FORMULA		7	1	Message-call into an account.	Complex -- see yellow paper Appendix H	
0xf2	CALLCODE	FORMULA		7	1	Message-call into this account with an alterna	Complex -- see yellow paper Appendix H	
0xf3	RETURN	0	zero	2	0	Halt execution returning output data.		
0xf4	DELEGATECALL	FORMULA		6	1	Message-call into this account with an alterna	Complex -- see yellow paper Appendix H	
0xfe	INVALID	NA		NA	NA	Designated invalid instruction.		
0xff	SELFDESTRUCT	FORMULA		1	0	Halt execution and register account for later d	5000 + ((create_	5000 for the operation plus 25000 if a new account is also created. A refund of 24000 gas is also added to the refund counter for self-destructing the account.