

0s: Stop and Arithmetic Operations		
0x00	STOP	Halts execution
0x01	ADD	Addition operation
0x02	MUL	Multiplication operation
0x03	SUB	Subtraction operation
0x04	DIV	Integer division operation
0x05	SDIV	Signed integer division operation (truncated)
0x06	MOD	Modulo remainder operation
0x07	SMOD	Signed Modulo remainder operation
0x08	ADDMOD	Modulo addition operation
0x09	MULMOD	Modulo multiplication operation
0x0a	EXP	Exponential operation
0x0b	SIGNEXTEND	Extend length of two's complement signed integer

50s: Stack, Memory, Storage and Flow Operations		
0x50	POP	Remove item from stack
0x51	MLOAD	Load word from memory
0x52	MSTORE	Save word to memory
0x53	MSTORE8	Save byte to memory
0x54	SLOAD	Load word from storage
0x55	SSTORE	Save word to storage
0x56	JUMP	Alter the program counter
0x57	JUMPI	Conditionally alter the program counter
0x58	PC	Get the value of the program counter prior to the increment corresponding to this instruction
0x59	MSIZE	Get the size of active memory in bytes
0x5a	GAS	Get the amount of available gas, including the corresponding reduction

50s: Stack, Memory, Storage and Flow Operations (cont)		
0x5b	JUMPDEST	Mark a valid destination for jumps

### 10s: Comparison & Bitwise Logic Operations

0x10	LT	Less-than comparison
0x11	GT	Greater-than comparison
0x12	SLT	Signed less-than comparison
0x13	SGT	Signed greater-than comparison
0x14	EQ	Equality comparison
0x15	ISZERO	Simple not operator
0x16	AND	Bitwise AND operation
0x17	OR	Bitwise OR operation
0x18	XOR	Bitwise XOR operation
0x19	NOT	Bitwise NOT operation
0x1a	BYTE	Retrieve single byte from word

20s: SHA3		
0x20	SHA3	Compute Keccak-256 hash

### 60s & 70s: Push Operations

0x60	PUSH1	Place 1-byte item on stack
0x61	PUSH2	Place 2-byte item on stack
0x7f	PUSH32	Place 32-byte (full word) item on stack

### 80s: Duplication Operations

0x80	DUP1	Duplicate 1st stack item
0x81	DUP2	Duplicate 2nd stack item
0x8f	DUP16	Duplicate 16th stack item

### 90s: Exchange Operations

0x90	SWAP1	Exchange 1st and 2nd stack items
0x91	SWAP2	Exchange 1st and 3rd stack items
0x9f	SWAP16	Exchange 1st and 17th stack items



30s: Environmental Information			30s: Environmental Information (cont)			f0s: System operations			40s: Block Information		
0x30	ADDRESS	Get address of currently executing account	0x39	CODECOPY	Copy code running in current environment to memory	0xf0	CREATE	Create a new account with associated code	0x40	BLOCKHASH	Get the hash of one of the 256 most recent complete blocks
0x31	BALANCE	Get balance of the given account	0x3a	GASPRICE	Get price of gas in current environment	0xf1	CALL	Message-call into an account	0x41	COINBASE	Get the block's beneficiary address
0x32	ORIGIN	Get execution origination address	0x3b	EXTCODESIZE	Get size of an account's code	0xf2	CALLCODE	Message-call into this account with an alternative account's code	0x42	TIMESTAMP	Get the block's timestamp
0x33	CALLER	Get caller address	0x3c	EXTCODECOPY	Copy an account's code to memory	0xf3	RETURN	Halt execution returning output data	0x43	NUMBER	Get the block's number
0x34	CALLVALUE	Get deposited value by the instruction/transaction responsible for this execution	<b>a0s: Logging Operations</b>			0xf4	DELEGATECALL	Message-call into this account with an alternative account's code, but persisting the current values for sender and value	0x44	DIFFICULTY	Get the block's difficulty
0x35	CALLDATASIZE	Get input data of current environment	0xa0	LOG0	Append log record with no topics	0xfe	INVALID	Designated invalid instruction	0x45	GASLIMIT	Get the block's gas limit
0x36	CALLDATASIZE	Get size of input data in current environment	0xa1	LOG1	Append log record with one topic	0xff	SELFDESTRUCT	Halt execution and register account for later deletion			
0x37	CALLDATACOPY	Copy input data in current environment to memory	0xa2	LOG2	Append log record with two topics						
0x38	CODESIZE	Get size of code running in current environment	0xa3	LOG3	Append log record with three topics						
			0xa4	LOG4	Append log record with four topics						

