

Ethereum Blockchain Mechanism (Proof Of Work)

An interpretation of the Ethereum Project Yellow Paper

G. Wood, "Ethereum: A secure decentralised generalised transaction ledger", 2014.

Lee Thomas

2016-06-21

Ver 0.1 2016-06-22

Successful miner's computer

Takes the following steps and broadcasts block header, H , to network

Determine Transactions

Miner picks transactions to process (from those broadcasted)

Determine Ommers

Miner finds and includes valid ommers

Apply Rewards

Update account balance(s) to reward valid blocks

Compute a Valid State

Block Finalisation Defines result of all selected state transitions

State Transition Cycle Defines result of a single transaction $\sigma_{t+1} = Y(\sigma_t, T)$

Execution Cycle Defines result of a single cycle of the state machine

Ethereum Virtual Machine, EVM

Instruction Set

0x00 STOP
0x01 ADD
0x02 MUL
0x03 SUB
0x04 DIV
...

Execution Environment, I

Code owner, I_c
Address of accounts that owns executing code

sender, I_s
Sender address of transaction that originated this execution

Gas price, I_g
Price of gas in the transaction that originated this execution

Input data, I_d
First array (transaction data if execution is in transaction)

causer, I_c
Address of account that caused the code to be executing (I_c if transaction)

value, I_v
Value sent to this account as part of execution procedure (transaction value if execution is in transaction)

Machine code, I_m
Hash header of the current block

Block header, I_H
Hash header of the current block

Message-call depth, I_d
Number of calls to CALL or CALLCODE being executed in present

Substate, A
Tuple

Suicide Set, A_s
Addresses to be deleted post transaction

Log Series, A_l
A series of structured (structured) data (allow contract calls to be easily tracked externally)

Refund Balance, A_r
Price of gas in the transaction that originated this execution

World State, σ

Addresses
under the number of transactions that have been added to, or the state of accounts with associated code, the number of contracts created and by the account

Account States
RLP encoded state accounts

Balances
under the number of Wei owned by this address

Storage and Code

Machine state, μ

Gas available, g
255 max

Program counter, pc
255 max

Memory contents, m

No words, i
Address number of words in memory, counting from position 0

Stack contents, s

Iterator Function, O
Defines result of a single cycle of the state machine

Get next instruction from I_b

Get items to add to /remove from stack, $\Delta = \alpha + \delta$

Update machine stack

Subtract gas used

Increment Program counter

Halt?

Yes

No

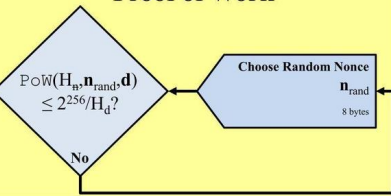
All transactions?

Yes

No

Get next Transaction

Proof of Work



Mining Network
"Oh look, some transactions have been broadcast to the network. Let's race each other to create a new valid block... GO"

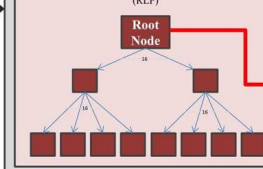


Red indicates KECCAK256 Hash

Information required to derive Block Header

Account storage contents Trie

A mapping between integer keys (KEC) and integer values (RLP)



Account, $\sigma[\text{address}]$

nonce, $\sigma[\text{address}]_n$
under the number of transactions that have been added to, or the state of accounts with associated code, the number of contracts created and by the account

balance, $\sigma[\text{address}]_b$
under the number of Wei owned by this address

storageRoot, $\sigma[\text{address}]_s$
The 32-bit hash of the root node of a Merkle Patricia tree that encodes the storage contents of the account's associated data. The storage contents are encoded into the tree as a mapping from the account's 256-bit hash to the 256-bit storage key in the RLP-encoded data

codeHash, $\sigma[\text{address}]_c$
The 32-bit hash of the code associated with the account. The code is encoded into the tree as a mapping from the account's 256-bit hash to the 256-bit code hash in the RLP-encoded data

Transaction, T

nonce, T_n
under the number of transactions that have been added to, or the state of accounts with associated code, the number of contracts created and by the account

gasPrice, T_g
under the number of Wei to be paid per unit of gas for all computations. This is the amount of gas that should be used in executing this transaction. The value is encoded in the RLP-encoded data

gasLimit, T_l
under the maximum amount of gas that should be used in executing this transaction. The value is encoded in the RLP-encoded data

to, T_t
160-bit address of the message call's recipient or, in the context of a contract transaction, the address of the contract

value, T_v
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

init, T_i
Contract creation information only and contains data only when creating a new contract. The value is encoded in the RLP-encoded data

data, T_d
Message call data only and contains data only when creating a new contract. The value is encoded in the RLP-encoded data

$V, T, S, T_n, T_g, T_l, T_t, T_v, T_i, T_d$

Transaction Receipt, $B_R[i]$
(i =transaction no)

post-transaction state, R_p
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

cumulative gas used, R_u
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

transaction logs, R_l
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

bloom filter of log info, R_b
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

Log Entry, O
Tuple

Logger's address, O_a
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

Log topics, O_t
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

Log data, O_d
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

Transaction Receipts Trie
Index keyed trie

Root Node

16

16

16

16

16

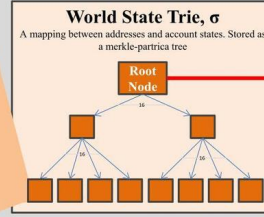
16

16

16

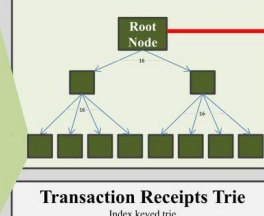
World State Trie, σ

A mapping between addresses and account states. Stored as a merkle-patricia tree



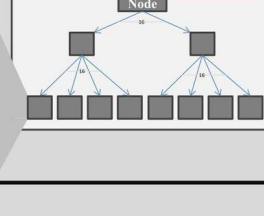
Transaction Trie, T

A merkle-patricia tree of transactions to include



Transaction Receipts Trie

Index keyed trie



Block, B

Block Header, H or B_H

parentHash, H_p
Keccak-256 hash of previous block's header

ommersHash, H_o
Keccak-256 hash of the ommers list of the previous block

beneficiary, H_b
160-bit address of the beneficiary of the transaction

stateRoot, H_s
Keccak-256 hash of the root node of the state trie, after all transactions in the block have been applied

transactionsRoot, H_t
Keccak-256 hash of the root node of the transaction trie

receiptsRoot, H_r
Keccak-256 hash of the root node of the receipt trie

logsBloom, H_l
Bloom filter from all logs in the block. A log is a topic associated with a transaction. The value is encoded in the RLP-encoded data

difficulty, H_d
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

number, H_n
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

gasLimit, H_g
under the maximum amount of gas that should be used in executing this transaction. The value is encoded in the RLP-encoded data

gasUsed, H_u
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

timestamp, H_t
under the number of Wei to be transferred in the message call's recipient or, in the context of a contract transaction, the address of the contract

extraData, H_e
arbitrary data up to 32 bytes. The value is encoded in the RLP-encoded data

mixHash, H_m
Keccak-256 hash which provides additional data to the state trie that is sufficient to reconstruct the state trie

nonce, H_n
Keccak-256 hash which provides additional data to the state trie that is sufficient to reconstruct the state trie

Transaction List, B_T

Ommers List, B_U