BitcoinBT (BTCBT)

Whitepaper (English Version)

---

1. Introduction

BitcoinBT (BTCBT) is a Proof-of-Work (PoW) blockchain that follows a hard-fork structure inheriting the existing Bitcoin chain up to block height 903,844.

This project does not aim to design a new consensus model. Instead, it originated from a structural review of whether limited consensus parameters can be adjusted while preserving the core architecture of an existing PoW chain.

BitcoinBT does not create a new genesis block.
All block data prior to the fork point remains unchanged, and the chain continues from that inherited state. New consensus conditions apply starting from the defined post-fork block height.

The consensus transition is automatically applied at a explicitly defined block height. No external approval process or centralized control mechanism is involved.

The key parameters applied after the fork are as follows:

- Fork reference block: 903,844 (full inheritance of prior history)
- Consensus activation height: 903,845 (new rules apply from this block onward)
- Target block interval: 300 seconds
- Difficulty adjustment: Per-block adjustment based on ASERT
- Consensus maximum block size: 32 MB (32,000,000 bytes)
- Proof-of-Work method: SHA-256 (double SHA-256)
- Halving interval: 210,000 blocks (approximately 2 years at a 300-second target)
- Maximum supply cap (Production Mainnet): 21,000,000 BTCBT

The maximum supply cap on the Production Mainnet is defined as 21,000,000 BTCBT.

A special reward of 630,000 BTCBT is defined within this cap and does not represent an expansion of total supply. This special reward is issued once at block height 903,850.

The issuance structure is determined by consensus rules. Blocks that include a reward exceeding the defined maximum supply are not recognized as valid blocks.

BitcoinBT maintains the original PoW structure, block header format, transaction validation model, UTXO state model, and accumulated-work-based chain selection rule. Chain selection is always determined by accumulated work.

BitcoinBT is implemented as a consensus-fork structure based on the Bitcoin Core v26 codebase. The application of consensus rules is determined automatically by block height. There are no external approval mechanisms, voting procedures, or signal-based activation schemes.

This whitepaper provides an overview of BitcoinBT's design architecture, scope of consensus definitions, network model, economic structure, and current project phase. Detailed code-level implementation aspects are addressed in separate technical documentation. The ultimate reference for consensus rules is the publicly released source code deployed on the Production Mainnet.

The Test Mainnet operates as a separate environment for functional and performance validation, and experimental parameters may be applied within that environment.

2. Background and Design Basis

BitcoinBT originated from an engineering review of whether a limited set of consensus definitions can be adjusted while preserving the structural continuity of an existing Proof-of-Work blockchain.

The objective was not to redesign the core consensus model, but to assess whether specific consensus parameters could be modified without altering the fundamental architecture of the inherited chain.

From the outset, the following design constraints were established:

- The SHA-256 Proof-of-Work mechanism is not modified.
- The block header structure remains unchanged.
- The transaction structure remains unchanged.
- The UTXO-based state model remains unchanged.
- The accumulated-work-based chain selection rule is preserved.

- The inherited genesis block is not replaced or redefined.
- Signature verification follows the same validation model as Bitcoin.

BitcoinBT does not introduce a separate signature scheme such as a ForkID mechanism. Signature validation continues to follow the inherited rules of the underlying chain. These constraints are intended to maintain structural continuity.

The fork reference point is defined at block height 903,844.
All blocks prior to this height are inherited without modification.
From block height 903,845 onward, modified consensus parameters are applied.

Consensus branching is determined exclusively by block height.
Pre-fork blocks follow the inherited validation rules, while post-fork blocks follow the updated consensus parameters and calculation paths. This branching logic is embedded in the software and cannot be selectively disabled at runtime.

Consensus-level changes are limited to the following areas:

- Target block interval definition
- Difficulty adjustment mechanism (ASERT-based computation)
- Consensus maximum block size
- Block-height-based consensus branching logic

All other core structures remain unchanged, including:

- SHA-256 Proof-of-Work
- Block header format
- Transaction data structure
- UTXO state model
- Accumulated-work-based chain selection rule

BitcoinBT therefore represents a parameter-level consensus adjustment within a preserved PoW architecture, rather than a structural redesign of the base protocol.

3. Design Principles and Scope of Consensus Changes

BitcoinBT is designed on the premise that the structural foundation of the inherited

Proof-of-Work blockchain remains intact. Consensus changes are deliberately limited in scope, and the underlying architecture of the protocol is not redesigned.

3.1 Structural Continuity

BitcoinBT inherits the complete blockchain history up to the defined fork height.
Validation rules for pre-fork blocks remain unchanged.
Post-fork consensus parameters apply only to blocks with height equal to or greater than the activation height.

This ensures that historical data integrity is preserved while enabling clearly defined forward modifications.

3.2 Limited Scope of Consensus Adjustments

Consensus modifications are restricted to a specific set of parameters:

- Target block interval definition
- Difficulty adjustment algorithm (ASERT-based computation)
- Consensus maximum block size

The following components remain unchanged:

- SHA-256 Proof-of-Work mechanism
- Block header structure
- Transaction data structure
- UTXO-based state model
- Accumulated-work-based chain selection rule

By limiting the scope of modification, BitcoinBT maintains compatibility with the inherited structural model while introducing defined parameter-level changes.

3.3 Block-Height-Based Consensus Branching

Consensus branching is determined exclusively by block height.
Blocks below the fork height are validated under inherited rules.
Blocks at or above the activation height are validated under the updated consensus conditions.

This branching logic is embedded in the validation flow of the software and cannot be toggled through runtime configuration.

3.4 Deterministic Validation Behavior

Nodes running identical software produce identical validation outcomes for identical block inputs.
Consensus application does not depend on signaling mechanisms, voting schemes, or centralized coordination.

Consensus changes occur only through software updates voluntarily adopted by network participants.

4. Block Interval and Difficulty Adjustment

BitcoinBT defines a target block interval of 300 seconds.
This parameter represents the intended long-term average block production interval within consensus.

Actual block time may vary depending on network hash rate conditions. The difficulty adjustment mechanism is designed to align the long-run average block interval with the defined target.

4.1 Target Block Interval

The target block interval is set to 300 seconds.

Since issuance and halving are defined in block counts rather than elapsed time, maintaining a 210,000-block halving interval results in a shorter time-based cycle compared to a 600-second target interval.

This parameter affects the temporal distribution of issuance but does not change the block-count-based emission structure.

4.2 Difficulty Adjustment Method

Post-fork difficulty adjustment is computed using an ASERT-based algorithm.
Difficulty is recalculated per block rather than at fixed periodic retarget intervals.

The ASERT half-life parameter is defined as 172,800 seconds (2 days).

Difficulty adjustment remains active at all times under post-fork conditions.
No minimum-difficulty exception rules are enabled in Production Mainnet consensus.

4.3 Scope of Application

The ASERT computation path applies only to blocks at or above the defined activation height.

Pre-fork blocks follow the inherited difficulty adjustment mechanism.
Post-fork blocks use the defined per-block adjustment logic.

This separation is determined deterministically by block height.

4.4 Validation Considerations

In validation environments, the following behaviors are confirmed:

- Per-block difficulty recalculation under varying time intervals
- Adjustment responsiveness under fluctuating block production rates
- Stability across fork-boundary transitions

Difficulty values are validated as part of block header verification.
Blocks that do not meet the defined target requirements are rejected during validation.

5. Economic Structure and Emission Model

BitcoinBT's monetary issuance is defined exclusively through block subsidies determined by consensus rules.
All nodes compute issuance amounts deterministically based on block height.

No dynamic monetary policy, external adjustment, or discretionary issuance mechanism exists within consensus.

5.1 Block-Subsidy-Based Issuance

New BTCBT units are created as block subsidies when valid blocks are added to the chain.

The subsidy amount is determined by block height and encoded in consensus logic.
Transaction fees are added to the subsidy and awarded to the block producer.

Issuance occurs only through valid block creation.

5.2 Halving Structure

The block subsidy is reduced by 50% every 210,000 blocks.

Halving is defined strictly by block count, not elapsed time.
With a 300-second target block interval, a 210,000-block cycle corresponds to approximately two years under long-run equilibrium conditions.

The halving schedule affects subsidy size but does not alter the consensus validation structure.

5.3 Maximum Supply Cap

The Production Mainnet maximum supply is defined as 21,000,000 BTCBT.

Subsidies decrease geometrically according to the halving schedule.
The total cumulative issuance converges toward the defined maximum supply cap.

A special reward of 630,000 BTCBT is issued once at block height 903,850.
This amount is included within the 21,000,000 BTCBT maximum supply and does not represent additional inflation beyond the defined cap.

Blocks containing rewards that would cause total issuance to exceed the consensus-defined maximum are not considered valid.

5.4 Miner Reward Composition

Block producers receive:

- Block subsidy
- Transaction fees included in the block

Reward calculation follows consensus rules and is validated by all nodes during block verification.

Any modification to issuance logic would require a consensus software update adopted by network participants.

6. Network Architecture and Participation Model

BitcoinBT operates as an open peer-to-peer (P2P) network.
Participation is achieved by running node software that independently validates and propagates blocks and transactions.

All nodes apply the same consensus rules to determine validity.
Invalid blocks or transactions are not incorporated into the local chain state.

6.1 Node Responsibilities

Nodes perform the following core functions:

- Receiving and validating transactions
- Receiving and validating blocks
- Propagating valid data to peers
- Maintaining and synchronizing the chain state

Validation results are determined exclusively by consensus rules encoded in the software.

6.2 Mining Participation

Block production follows the SHA-256 Proof-of-Work model.

The block creation process consists of:

1. Selecting valid transactions from the mempool
2. Constructing a candidate block header and coinbase transaction
3. Performing double SHA-256 hashing against the defined difficulty target
4. Broadcasting the block when a valid hash is found

Blocks are accepted by the network only after full validation by participating nodes.

6.3 Chain Selection Rule

Chain selection is based on accumulated work.

Nodes maintain the valid chain with the greatest cumulative Proof-of-Work.
Temporary forks may occur under network latency conditions, but resolution follows the accumulated-work rule deterministically.

6.4 Network Validation Behavior

Within operational and validation environments, the following behaviors are confirmed:

- Multi-node synchronization
- Block propagation across peers
- Deterministic rejection of consensus-invalid blocks
- Stable chain state updates during reorganizations

BitcoinBT does not introduce alternative finality mechanisms or centralized coordination layers.

7. Development Structure and Change Control

BitcoinBT is implemented as a consensus fork based on the Bitcoin Core v26 codebase. Modifications are limited to explicitly defined consensus parameters and the associated calculation paths required to apply them.

The underlying validation framework, transaction processing structure, and state management logic remain consistent with the inherited architecture.

7.1 Scope of Modifications

Consensus-related modifications include:

- Target block interval definition
- ASERT-based per-block difficulty computation
- Consensus maximum block size (32 MB)
- Block-height-based consensus branching logic
- Block subsidy definition aligned with the defined supply cap

No structural redesign of the block format, transaction structure, or UTXO model has been introduced.

7.2 Height-Based Activation Logic

Consensus changes are enforced through explicit conditional branching based on block height.

Pre-fork blocks are validated under inherited rules.
Post-fork blocks are validated under the updated consensus parameters.

This logic is embedded within the validation path of the node software and is not controlled through runtime configuration settings.

7.3 Consensus and Policy Separation

Consensus rules define block validity and chain state transitions.

Policy rules govern transaction relay behavior and mempool acceptance but do not alter block validity criteria.

BitcoinBT's modifications are confined to consensus-level definitions.
The policy layer remains structurally aligned with inherited behavior.

7.4 Deterministic Consensus Execution

Consensus validation depends solely on block data and chain state.

Nodes operating identical software under identical chain conditions will derive identical validation results.

Future consensus changes would require a coordinated software update adopted by network participants.

8. Security Model and Operational Stability

BitcoinBT follows a Proof-of-Work consensus model in which block validity and chain progression are determined by cryptographic hashing and accumulated work.

Network security is influenced by hash rate distribution, block propagation conditions, and participation of independent validating nodes.

8.1 Proof-of-Work Structure

Blocks are produced by performing double SHA-256 hashing on the block header.

A block is considered valid only if its hash value is less than or equal to the consensus-defined target.

All nodes independently verify this condition during block validation.

Chain legitimacy is determined by accumulated Proof-of-Work rather than by identity, authority, or external coordination.

8.2 Implications of a 300-Second Target Interval

A 300-second target interval increases the number of blocks produced per unit time compared to a 600-second interval.

This affects:

- The frequency of subsidy issuance
- The relative impact of propagation latency

- The responsiveness of difficulty adjustment

Short-term chain reorganizations may occur under normal network latency conditions, as in other PoW systems. Resolution follows accumulated-work rules.

8.3 Difficulty Adjustment and Stability

Post-fork difficulty is recalculated per block using an ASERT-based adjustment mechanism.

If block production deviates from the defined target interval, the difficulty target adjusts according to the algorithm to maintain long-run equilibrium around the 300-second average.

There is no manual intervention mechanism in the difficulty computation process.

8.4 Block Size Cap and Validation Load

The consensus maximum block size is defined as 32 MB.

Blocks exceeding this size are invalid under consensus rules.

Larger blocks may increase validation and bandwidth requirements. Block validity is determined strictly by defined structural and script verification rules.

8.5 Operational Validation

In validation and operational environments, the following behaviors are confirmed:

- Deterministic rejection of consensus-invalid blocks
- Stable chain progression under normal network conditions
- Proper chain reorganization handling
- Consistent validation results across multiple nodes

BitcoinBT does not introduce alternative finality layers or centralized override mechanisms.

9. Current Status and Project Phase

BitcoinBT is currently in an implementation and validation phase focused on confirming that the defined consensus parameters are correctly reflected in observable chain behavior.

The objective of this phase is to verify that the intended consensus definitions operate deterministically under real execution conditions.

9.1 Verified Areas

In post-fork regions, the following elements have been validated in controlled environments:

- Application of the 300-second target block interval
- Proper execution of ASERT-based per-block difficulty computation
- Enforcement of the 32 MB consensus maximum block size
- Deterministic block subsidy calculation according to defined supply rules

Validation focuses on structural correctness rather than performance optimization.

9.2 Chain Progress and Block Production

Test environments have confirmed:

- Successful block production under post-fork rules
- Consistent block height progression
- Correct application of issuance logic
- Header validation behavior consistent with consensus definitions

Only blocks satisfying all validation conditions are connected to the chain state.

9.3 Code-Based Consensus Application

Consensus definitions are embedded in the source code and activated via block-height-based conditional branching.

Consensus parameters are defined explicitly and are not modified dynamically at runtime.

Any future change to consensus definitions would require a new software release adopted by network participants.

9.4 Scope Limitation

This document does not provide large-scale production network metrics, third-party audits, or long-term operational forecasts.

The purpose of this section is limited to describing the current implementation and validation status.

10. Scope and Closing Statement

This document provides a structural overview of BitcoinBT's consensus framework, economic design, and network model.

BitcoinBT preserves the fundamental architecture of an inherited Proof-of-Work blockchain while applying explicitly defined parameter-level adjustments at a predetermined block height.

The scope of this whitepaper includes:

- Block-height-based consensus transition structure
- 300-second target block interval
- ASERT-based difficulty adjustment model
- 32 MB consensus maximum block size
- 21,000,000 BTCBT maximum supply cap
- Block subsidy and halving structure

The document is limited to describing defined consensus parameters and implementation scope.

This whitepaper does not constitute investment solicitation, financial advice, legal guidance, or market projection.

The authoritative reference for consensus rules is the publicly released source code executed on the Production Mainnet.

Test Mainnet environments may apply experimental configurations for validation purposes and do not supersede Production Mainnet definitions.