BitcoinBT (BTCBT)

Whitepaper

---

1. Introduction

BitcoinBT (BTCBT) is a Proof-of-Work (PoW) blockchain implemented as a consensus hard fork, inheriting the Bitcoin blockchain history up to block height 903,844.

The project does not seek to introduce a new consensus model. Rather, it originated from a structural review assessing whether limited consensus parameters could 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 beginning at a defined post-fork block height.

The consensus transition is enforced automatically at an explicitly defined block height. No external approval process, centralized control mechanism, voting procedure, or signaling-based activation scheme is involved.

The key parameters applied after the fork include:

- Fork point (inherited history up to): 903,844
- Consensus activation height: 903,845
- 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-256d (double SHA-256)
- Halving interval: 210,000 blocks (approximately 729 days at a 300-second target interval)
- Maximum supply cap (BitcoinBT Mainnet): 21,000,000 BTCBT

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

A one-time special reward of 630,000 BTCBT is issued at block height 903,850 and is included within the 21,000,000 BTCBT maximum supply cap; it does not increase the cap.

Issuance is determined strictly by consensus rules. A block is considered invalid if its coinbase subsidy violates the consensus-defined issuance limits, including the maximum supply cap.

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

BitcoinBT is implemented as a consensus fork based on the Bitcoin Core v26 codebase. Consensus application is determined automatically by block height.

This whitepaper provides an overview of BitcoinBT's design architecture, defined consensus scope, network model, economic structure, and current project phase. Detailed implementation specifics are addressed in separate technical documentation. The authoritative reference for consensus rules is the publicly released source code executed by BitcoinBT Mainnet nodes.

The Test Network operates as a separate, non-production environment for functional and performance validation.
It does not define or modify BitcoinBT Mainnet consensus rules. Non-mainnet parameters may be applied in that environment strictly for testing purposes.

2. Background and Design Basis

BitcoinBT originated from an engineering review assessing whether a limited set of consensus parameters could 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 determine 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 remains unchanged.
- The block header structure remains unchanged.
- The transaction data 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 an alternative signature scheme, such as a ForkID mechanism. Signature validation continues to follow the inherited verification rules. These constraints are intended to preserve structural consistency with the underlying architecture.

The fork point is defined at block height 903,844.
All blocks below this height are inherited without modification.
From block height 903,845 onward, the defined consensus adjustments are applied.

Consensus branching is determined exclusively by block height.
Pre-fork blocks are validated under inherited rules, while post-fork blocks follow the updated consensus parameters and calculation paths. This branching logic is embedded directly in the validation flow of the software and cannot be altered through runtime configuration.

Consensus-level changes are confined to the following areas:

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

All other core components remain unchanged, including:

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

BitcoinBT therefore represents a parameter-level consensus refinement within a preserved PoW framework, 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 intentionally limited in scope, and the fundamental protocol architecture is not redesigned.

3.1 Structural Continuity

BitcoinBT inherits the full blockchain history up to the defined fork height.

Validation rules for pre-fork blocks are preserved.
Post-fork consensus parameters apply only to blocks with a height greater than or equal to the activation height.

This approach preserves historical integrity while allowing explicitly defined forward adjustments.

3.2 Limited Scope of Consensus Adjustments

Consensus modifications are confined to a defined set of parameters:

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

The following structural components remain unchanged:

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

By restricting modifications to clearly defined parameters, BitcoinBT preserves the inherited structural model while applying explicitly defined consensus parameter adjustments.

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 follow the updated consensus conditions.

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

3.4 Deterministic Validation Behavior

Nodes running identical software produce identical validation results for identical block inputs.

Consensus application does not rely on signaling, voting mechanisms, or centralized coordination.
Any modification to consensus rules requires a software update adopted by network participants.

4. Block Interval and Difficulty Adjustment

BitcoinBT defines a target block interval of 300 seconds.

This parameter establishes the intended long-run average rate of block production within consensus. While actual block intervals may fluctuate due to variations in network hash rate, the difficulty adjustment mechanism deterministically adjusts the difficulty target to restore convergence toward the defined 300-second long-run average.

4.1 Target Block Interval

The protocol specifies a 300-second block interval target.

Because issuance and halving are defined by block count rather than elapsed time, maintaining a 210,000-block halving interval results in a shorter time-based emission cycle than a 600-second target interval. Under the 300-second target assumption, a 210,000-block cycle corresponds to approximately 729 days.

This adjustment affects the temporal distribution of issuance while preserving the block-count-based emission structure.

4.2 Difficulty Adjustment Method

Post-fork difficulty is calculated using an ASERT-based per-block adjustment algorithm.

Difficulty is recalculated for each block based on elapsed time deviation from the defined target interval, rather than at fixed periodic retarget intervals.

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

Difficulty adjustment remains continuously active under post-fork conditions. No minimum-difficulty exception rules are enabled under BitcoinBT Mainnet consensus.

4.3 Scope of Application

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

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

This separation is enforced deterministically through block-height-based validation branching.

4.4 Validation Considerations

In controlled validation environments, the following behaviors have been verified:

- Per-block difficulty recalculation under variable time intervals
- Responsiveness of difficulty adjustment under fluctuating block production rates
- Stability across fork-boundary transitions

Difficulty values form part of block header validation. Blocks that do not satisfy the consensus-defined target requirements are considered invalid.

5. Economic Structure and Emission Model (Revised)

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 discretionary monetary policy, external adjustment mechanism, or dynamic issuance control exists within consensus.

5.1 Block-Subsidy-Based Issuance

New BTCBT units are created solely through block subsidies upon validation and connection of new blocks to the chain.

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

Issuance occurs exclusively through valid block creation under consensus rules.

5.2 Halving Structure

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

Halving is defined strictly by block count rather than elapsed time.
With a 300-second target block interval, a 210,000-block halving cycle corresponds to approximately 729 days under long-run equilibrium conditions, making the time-based cycle shorter than under a 600-second target interval.

The halving schedule affects subsidy magnitude but does not modify the structural validation logic of the protocol.

5.3 Maximum Supply Cap

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

Block subsidies decrease geometrically according to the defined halving schedule.
As a result, cumulative issuance asymptotically converges toward the maximum supply cap.

A special reward of 630,000 BTCBT is issued once at block height 903,850.
This reward is included within the 21,000,000 BTCBT cap and does not constitute

additional supply beyond the defined maximum.

Any block whose coinbase reward would cause cumulative issuance to exceed the 21,000,000 BTCBT cap is invalid under consensus rules.

5.4 Miner Reward Composition

Block producers receive:

- Block subsidy
- Transaction fees included in the block

Reward calculation is governed by consensus rules and independently verified by all validating nodes.

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

6. Network Architecture and Participation Model

BitcoinBT operates as an open peer-to-peer (P2P) network.

Network participation occurs through independently operated node software that validates and propagates blocks and transactions according to defined consensus rules.

All nodes apply identical validation logic. Blocks or transactions that fail to meet consensus requirements are rejected and 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 connected peers
- Maintaining and synchronizing the canonical chain state

Validation outcomes are determined exclusively by consensus logic embedded in the software.

6.2 Mining Participation

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

The block construction process includes:

1. Selecting valid transactions from the mempool
2. Constructing a candidate block, including the coinbase transaction
3. Performing double SHA-256 hashing against the consensus-defined difficulty target
4. Broadcasting the block once a valid hash meeting the target requirement is discovered

A block is connected to the chain only after full validation by participating nodes across the network.

6.3 Chain Selection Rule

Chain selection is determined by cumulative Proof-of-Work.

Nodes adopt and extend the valid chain that represents the greatest accumulated computational work.

Temporary chain splits may occur under normal network latency conditions; however, resolution follows the accumulated-work rule deterministically and does not rely on centralized coordination.

6.4 Network Validation Behavior

In operational and validation environments, the following behaviors have been observed:

- Multi-node synchronization consistency
- Reliable block propagation across peers

- Deterministic rejection of consensus-invalid blocks
- Stable chain state updates during reorganizations

BitcoinBT does not implement alternative finality mechanisms, checkpoint overrides, or centralized coordination layers.

7. Development Structure and Change Control

BitcoinBT is implemented as a consensus fork derived from the Bitcoin Core v26 codebase.

Modifications are confined to explicitly defined consensus parameters and the calculation paths required to enforce them. The core validation engine, transaction processing logic, and state management framework remain aligned with the inherited architecture.

7.1 Scope of Modifications

Consensus-level modifications include:

- Target block interval definition
- ASERT-based per-block difficulty adjustment
- 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, scripting engine, 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 defined consensus adjustments.

This branching logic is embedded directly within the validation pipeline and cannot be modified through runtime configuration.

7.3 Consensus and Policy Separation

Consensus rules determine block validity and state transitions.

Policy rules govern transaction relay and mempool behavior but do not affect consensus validity criteria.

BitcoinBT's modifications are strictly limited to consensus-level definitions.
The policy layer retains structural compatibility with inherited behavior.

7.4 Deterministic Consensus Execution

Consensus validation depends solely on block data and chain state.

Nodes executing identical software under identical chain conditions will derive identical validation outcomes.

Any future modification to consensus rules would require a coordinated software release adopted by network participants.

---

8. Security Model and Operational Stability

BitcoinBT follows a Proof-of-Work (PoW) consensus model in which block validity and chain progression are determined by cryptographic hashing and cumulative computational work.

Network security depends on hash rate distribution, block propagation latency, and independent participation of validating nodes.

8.1 Proof-of-Work Structure

Blocks are produced through double SHA-256 hashing of the block header.

A block is valid only if its hash satisfies the consensus-defined difficulty target.

All nodes independently verify this requirement during block validation.

Chain legitimacy is determined exclusively by cumulative Proof-of-Work rather than by identity, authority, or centralized coordination.

8.2 Implications of a 300-Second Target Interval

A 300-second target interval increases block frequency relative to a 600-second interval.

This influences:

- The temporal distribution of subsidy issuance
- Sensitivity to propagation latency
- The responsiveness of difficulty adjustment

Short-term chain reorganizations may occur under normal network conditions. Resolution follows the cumulative-work rule deterministically.

8.3 Difficulty Adjustment and Stability

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

When observed block intervals deviate from the defined target, the difficulty target adjusts according to the algorithm to maintain convergence toward the long-run 300-second average.

The difficulty computation process does not permit manual override or discretionary intervention.

8.4 Block Size Cap and Validation Load

The consensus maximum block size is defined as 32 MB.

Blocks exceeding this limit are invalid under consensus rules.

Larger blocks may increase bandwidth and validation resource requirements; however, block validity is determined strictly by consensus-defined structural and script verification rules.

8.5 Operational Validation

Operational validation environments have confirmed:

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

BitcoinBT does not implement alternative finality layers, mandatory checkpoint overrides, centralized arbitration, or externally enforced rollback mechanisms.

9. Current Status and Project Phase

BitcoinBT is currently in an implementation and validation phase.

The purpose of this phase is to confirm that the defined consensus parameters are correctly enforced by the node software and that resulting chain behavior is deterministic under execution conditions consistent with network operation.

9.1 Validated Elements

Within post-fork regions, the following components have been verified in controlled validation environments:

- Enforcement of the 300-second target block interval through difficulty adjustment behavior
- Correct execution of ASERT-based per-block difficulty computation
- Enforcement of the 32 MB consensus maximum block size
- Deterministic block subsidy calculation aligned with the defined supply cap

Validation activities focus on consensus correctness and rule enforcement rather than throughput benchmarking or performance optimization.

9.2 Chain Progression and Block Production

Validation environments have confirmed:

- Successful block production under post-fork consensus rules
- Deterministic block height progression
- Correct application of issuance logic in coinbase validation
- Block header verification consistent with consensus-defined difficulty targets

Blocks are connected to chain state only after full validation under defined consensus rules.

9.3 Code-Level Consensus Enforcement

Consensus parameters are defined explicitly within the source code.

Activation of post-fork rules occurs exclusively through block-height-based branching logic embedded in the validation path.

Consensus parameters are not dynamically adjustable at runtime.

Any modification to consensus definitions would require a formal software release adopted by network participants.

9.4 Scope Clarification

This document does not provide:

- Third-party security audits
- Economic projections
- Network performance guarantees
- Market or operational forecasts

This section is limited to describing implementation status and consensus validation scope.

---

10. Scope and Closing Statement

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

BitcoinBT preserves the core structure of the inherited Proof-of-Work blockchain while introducing explicitly defined parameter-level consensus adjustments at a predetermined block height.

10.1 Scope of Definition

This document covers:

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

All descriptions are limited to consensus-level definitions and implementation scope.

10.2 Authority of Reference

This whitepaper does not constitute:

- Investment solicitation
- Financial advice
- Legal interpretation
- Market forecast

The authoritative definition of consensus rules is the publicly released source code executed by BitcoinBT Mainnet nodes.

Test Network environments may apply experimental configurations for validation purposes. Such configurations do not supersede BitcoinBT Mainnet consensus definitions.