Quantstamp

Ethena UStb Token And Minting

Executive Summary

This audit report was prepared by Quantstamp, the leader in blockchain security.

Type	DeFi stablecoin
Timeline	2024-10-23 through 2024-10-25
Language	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	AUDIT.md
Source Code	• https://github.com/ethena-labs/ethena-ustb-audit #d82676f
Auditors	• Valerian Callens Senior Auditing Engineer • Roman Rohleder Senior Auditing Engineer • Rabib Islam Senior Auditing Engineer

Documentation quality	High
Test quality	Medium
Total Findings	5 Acknowledged: 5
High severity findings	0
Medium severity findings	0
Low severity findings	1 Acknowledged: 1
Undetermined severity findings	1 Acknowledged: 1
Informational findings	3 Acknowledged: 3

Summary of Findings

This project consists of 4 contracts to manage UStb tokens, a stablecoin backed by collateral. The system includes contracts for minting, redeeming, and managing UStb tokens with features such as whitelisting, blacklisting, and role-based access control. The contracts leverage OpenZeppelin libraries for security and utility functions.

Overall, the code is well-written, well-documented, and follows best practices. No major issue has been identified. The test suite is quite extensive but coverage results could be improved.

We highly appreciate that the Ethena team was responsive throughout the audit, promptly addressing our questions and participating in productive discussions.

** Fix review update **

The Ethena team addressed all issues in this report by acknowledging or fixing them. Several tests were added to cover the different types of transfers.

ID	DESCRIPTION	SEVERITY	STATUS
USTB-1	Missing Input Validations	Low	Acknowledged
USTB-2	Missing Storage Gaps in Inherited Contract	Informational	Acknowledged
USTB-3	Risks of Supporting Non-Standard ERC-20 Tokens	Informational	Acknowledged
USTB-4	Considerations about Events	Informational	Acknowledged
USTB-5	Depending on How Nonces Are Calculated Off-Chain, Nonce Verification May Reject Valid Nonces	Undetermined	Acknowledged

Assessment Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

> **Disclaimer**
> Only features that are contained within the repositories at the commit hashes specified on the front page of the report are within the scope of the audit and fix review. All features added in future revisions of the code are excluded from consideration in this report.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

1. Code review that includes the following
    1. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
    2. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
    3. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
    1. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
    2. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Scope

Files Included

Repo: https://github.com/ethena-labs/ethena-ustb-audit(d82676fa43cecab2832cc4804d029b4a07df408f)
Files:

- contracts/contracts/SingleAdminAccessControl.sol
- contracts/contracts/SingleAdminAccessControlUpgradeable.sol
- contracts/contracts/ustb/UStb.sol
- contracts/contracts/ustb/UStbMinting.sol

Files Excluded

Repo: https://github.com/ethena-labs/ethena-ustb-audit(d82676fa43cecab2832cc4804d029b4a07df408f)
Files: contracts/interfaces/, *contracts/libs/*, contracts/mocks/*

Operational Considerations

1. The contracts `UStb` and through inheritance `SingleAdminAccessControlUpgradeable` are upgradable and may have their logic modified in the future. Such potential changes were out-of-scope for this audit.
2. In `UStb`, there is a centralization of risks with the account with the role `DEFAULT_ADMIN_ROLE` with these two privileges:
   - `P_1`: can assign blacklisters that can blacklist users via `addBlacklistAddress()`;
   - `P_2`: can transfer tokens from blacklisted users via `redistributeLockedAmount()`;
   As a result, a malicious actor would just need to compromise the address with the role `DEFAULT_ADMIN_ROLE` to be able to use these two privileges together and transfer all minted `UStb` to a particular address (by using `P_1` and then `P_2` on all token holders). The roles owning `P_1` and `P_2` could be separated, such that a malicious actor would need to compromise two distinct accounts to transfer all

minted `UStb`. Compromising only one account would have a limited impact. It is advised to at least guard this role with i.e. multi-sigs to mitigate potential abuse.

3. The mutability of the address `UStb` in `UStbMinting` with the function `setUStbToken()` was confirmed to be an expected feature by the Ethena team. However, if such update happens, the following can happen:
   - users will not be able to redeem collateral because they own old `UStb` tokens since the call to `mint()`, and new `UStb` tokens should be burned when calling `redeem()`;
   - any already-signed non-expired orders would remain valid for the new `UStb` token since the address of `UStb` is not part of the struct `Order`;
4. There could be a race condition between concurrent actors for the sets of functions:
   - `mint()`, `redeem()`, `setGlobalMaxMintPerBlock()`, `setGlobalMaxRedeemPerBlock()`, `disableMintRedeem()`, `removeSupportedAsset()`, `removeMinterRole()`, `removeRedeemerRole()`, `addSupportedAsset()`, `setMaxMintPerBlock()` and `setMaxRedeemPerBlock()`;
   - `removeCustodianAddress()`, `removeCollateralManagerRole()`, `addCustodianAddress()` and `transferToCustody()`;
5. The last entry of the route addresses always has a slight advantage since any potential dust, after ratio computations, will be forwarded there.

Key Actors And Their Capabilities

In `SingleAdminAccessControl` and `SingleAdminAccessControlUpgradeable`:

1. The role `DEFAULT_ADMIN_ROLE` can execute:
   - `grantRole()` / `revokeRole()`: Grant/Revoke any address any role, except the `DEFAULT_ADMIN_ROLE` role.
   - `transferAdmin()`: Transfer the `DEFAULT_ADMIN_ROLE` role to another address in a 2-step process.

In `UStb`:

1. The role `DEFAULT_ADMIN_ROLE` can execute:
   - Same as `SingleAdminAccessControlUpgradeable`.
   - `addMinter()` / `setMinter()`: Add/Remove a minter contract.
   - `redistributeLockedAmount()`: Burn all tokens from a blacklisted address and mint the same amount to another non-blacklisted address.
   - `rescueTokens()`: Transfer any accidentally stuck ERC-20 tokens from the contract to another address.
   - `updateTransferState()`: Change the transfer restrictions:
     - `FULLY_ENABLED`: All transfers allowed, except from/to blacklisted addresses.
     - `WHITELIST_ENABLED`: All transfers disabled, except for whitelisted addresses.
     - `FULLY_DISABLED`: All transfers disabled.
2. The role `MINTER_CONTRACT` can execute:
   - `mint()`.
3. The role `BLACKLIST_MANAGER_ROLE` can execute:
   - `addBlacklistAddress()` / `removeBlacklistAddress()`: Add/Remove addresses to/from the blacklist (`BLACKLISTED_ROLE`), preventing/re-allowing them from transfers in the `FULLY_ENABLED` transfer mode and subjecting their tokens to be redistributed through the `DEFAULT_ADMIN_ROLE` role.
4. The role `WHITELIST_MANAGER_ROLE` can execute:
   - `addWhitelistAddress()` / `removeWhitelistAddress()`: Add/Remove addresses to/from the whitelist (`WHITELISTED_ROLE`), allowing/preventing them from transfers in the `WHITELIST_ENABLED` transfer mode.

In `UStbMinting`:

1. The role `DEFAULT_ADMIN_ROLE` can execute:
   - Same as `SingleAdminAccessControl`.
   - `setUStbToken()`: Change the `UStb` token address.
   - `setStablesDeltaLimit()`: Change the allowed stablecoin to collateral difference in BPS.
   - `setTokenType()`: Change the token type for an active collateral.
   - `setMaxRedeemPerBlock()`: Change the redeem cap per collateral, including setting it to zero or greater than the global cap.
   - `setMaxMintPerBlock()`: Change the mint cap per collateral, including setting it to zero or greater than the global cap.
   - `addSupportedAsset()` / `removeSupportedAsset()`: Add a new asset or remove an existing active one.
   - `addWhitelistedBenefactor()` / `removeWhitelistedBenefactor()`: Add/Remove addresses to/from the whitelisted benefactors list.
   - `addCustodianAddress()` / `removeCustodianAddress()`: Add/Remove addresses to/from the custodian list.
   - `setGlobalMaxRedeemPerBlock()`: Change the global redeem cap, including setting it to zero and thereby preventing future redeems.
   - `setGlobalMaxMintPerBlock()`: Change the global mint cap, including setting it to zero and thereby preventing future mints.
2. The role `GATEKEEPER_ROLE` can execute:
   - `renounceRole()`: Renounce the role and thereby lose the privilege to call any of the below listed function(s).
   - `removeCollateralManagerRole()`: Revoke the `COLLATERAL_MANAGER_ROLE` for a given address.
   - `removeRedeemerRole()`: Revoke the `REDEEMER_ROLE` for a given address.
   - `removeMinterRole()`: Revoke the `MINTER_ROLE` for a given address.
   - `disableMintRedeem()`: Prevent minting/redeeming by resetting the corresponding global caps back to zero.
3. The role `COLLATERAL_MANAGER_ROLE` can execute:
   - `renounceRole()`: See above.
   - `transferToCustody()`: Transfers an arbitrary asset to a whitelisted custodian address.
4. The role `REDEEMER_ROLE` can execute:
   - `renounceRole()`: See above.

- `redeem()` : Redeem the benefactors UStb tokens transfer signed collateral value to beneficiary, respecting the local and global block redeem cap.

5. The role `MINTER_ROLE` can execute:
   - `renounceRole()` : See above.
   - `mint()` : Transfer signed collateral amounts from benefactor to list of addresses with corresponding rations, respecting the local and global block mint cap and mints UStb tokens to the beneficiary.

Findings

USTB-1 Missing Input Validations

• Low ⓘ Acknowledged

Update
Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
1. Ack – won't fix
2. Ack – won't fix
3. Ack – uniqueness of nonce is checked in verifyNonce – won't fix
4. Ack – won't fix
```

File(s) affected: `UStb.sol, UStbMinting.sol`

Description: It is important to validate inputs, even if they only come from trusted addresses, to avoid human error:

1. In `UStb.sol`, function `initialize()`, the address `minterContract` may not be a contract;
2. In `UStb.sol`, function `addBlacklistAddress()`, it is possible to blacklist:
   - the minter contract and this could temporarily disrupt the mint and redeem operations;
   - the address with the role `DEFAULT_ADMIN_ROLE`, and this could temporarily disrupt the operation to redistribute locked amounts;
3. In `UStbMinting.sol`, function `verifyOrder()`, the uniqueness and the size of `Order.order_id` is not checked;
4. In `UStbMinting.sol`, function `setUStb()`, the value of `_ustb` is not checked to be:
   - Different from `address(0x0)`;
   - Different from existing supported assets;
   - Different from existing custodian;

Recommendation: Consider adding the relevant checks.

USTB-2 Missing Storage Gaps in Inherited Contract

• Informational ⓘ Acknowledged

Update
Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
won't fix
```

File(s) affected: `UStb.sol, SingleAdminAccessControlUpgradeable.sol`

Description: The `UStb` and `SingleAdminAccessControlUpgradeable` contracts are designed to be upgradable. Upgradable contracts usually have reserved space to allow future versions to add new state variables to the contract without shifting down storage in the inheritance chain. As `SingleAdminAccessControlUpgradeable` is inherited by `UStb`, adding a storage gap can be done to prevent storage collisions in future updates where storage slots could be added to `SingleAdminAccessControlUpgradeable`.

For instance, if in a future update, `SingleAdminAccessControlUpgradeable` is modified to include a new variable, and the `UStb` contract were upgraded to use the new version, this new variable would likely collide with the `transferState` variable in `UStb` (except in the special case where the new variable is given a custom slot).

Recommendation: Consider including a storage gap to the contract `SingleAdminAccessControlUpgradeable`. See OpenZeppelin's documentation for more details.

USTB-3 Risks of Supporting Non-Standard ERC-20 Tokens

• Informational ⓘ Acknowledged

Update
Marked as "Acknowledged" by the client.
The client provided the following explanation:

> `planned supported tokens don't have these features - won't fix`

File(s) affected: `UStbMinting.sol`

Description: Supporting tokens with specific features such as fees, rebasing, pausable, upgradeable, blacklist-able, or hooks on transfers could negatively impact the main flows of the system (deposits via `mint()`, transfer to custody, redeems via `redeem()`) if no specific mitigation measure is enforced to limit the consequences. For instance, in the function `_transferCollateral()` if `asset` represents an asset where the amount transferred is different than the amount requested to be transferred (ex: if fees are enforced), the actual amount transferred to custodians may differ from the expected transferred amount.

Recommendation: Consider analyzing thoroughly from a security perspective any token you would like to be supported.

USTB-4 Considerations about Events

Informational · Acknowledged

> **Update**
> Marked as "Acknowledged" by the client.
> The client provided the following explanation:
>
> `won't fix`

File(s) affected: `UStbMinting.sol`

Description:

1. The initial limits of a new asset added to `UStbMinting` (max mint and max redeem limits per block per asset) are not accessible to off-chain observers via events because the event `AssetAdded` has only one field: `AssetAdded(address indexed asset)`.
2. Updates made via the function `setStablesDeltaLimit()` are not logged.

Recommendation: Consider improving these items.

USTB-5 Depending on How Nonces Are Calculated Off-Chain, Nonce Verification May Reject Valid Nonces

Undetermined · Acknowledged

> **Update**
> Marked as "Acknowledged" by the client.
> The client provided the following explanation:
>
> `won't fix`

File(s) affected: `UStbMinting.sol`

Description: When orders are submitted to the functions `mint()` and `redeem()`, their field `uint128 nonce` is checked via the function `verifyNonce()`. That function uses the data structure `_orderBitmaps` to check if a sender already used or not a given nonce. In detail, it is a mapping storing for a given address a `uint256` bitmap called `invalidator` at the `uint128` keys called `invalidatorSlot`. However, the following line could lead to an issue:

```
uint128 invalidatorSlot = uint64(nonce) >> 8;
```

Casting `nonce` to `uint64` seems too restrictive. If any value of `uint128 nonce` is valid, it is possible to have two different values of `nonce` that will be stored at the same bit in the data structure `_orderBitmaps`. For instance, the two values: `nonce_a = (1 << 125) + (1 << 60)` and `nonce_b = (1 << 60)`, since any non-matching bit higher than the 64th bit will be ignored because of the cast `uint64(nonce)`. Ultimately, this could prevent valid Orders from being accepted by the contract.

Recommendation: Consider removing the cast operation to `uint64`, or make sure that the off-chain component does not provide nonces greater than `type(uint64).max`.

Auditor Suggestions

S1 Documentation Improvements

Fixed

Update

Marked as "Fixed" by the client.
Addressed in: `7368bb88321376538bce979b5556977dc63ed303`.
The client provided the following explanation:

```
3. Ack - won't fix
```

File(s) affected: `UStbMinting.sol, UStb.sol`

Description:

1. In `UStbMinting.sol`, there is a typo: `"/// @notice h2olds EIP712 revision;"`;
2. In `SingleAdminAccessControlUpgradeable .sol`, the comments in lines 9-10 still refer to this contract as `SingleAdminAccessControl` rather than `SingleAdminAccessControlUpgradeable`;
3. In `UStbMinting.addWhitelistedBenefactor()`: Missing NatSpec comment for parameter `benefactor`;
4. `UStbMinting.removeWhitelistedBenefactor()`: Missing NatSpec comment for parameter `benefactor`;
5. `UStbMinting.addCustodianAddress()`: Missing NatSpec comment for parameter `custodian`;
6. `UStbMinting.removeCustodianAddress()`: Missing NatSpec comment for parameter `custodian`;

Recommendation: Consider addressing the items listed above.

S2 Code Conciseness

Fixed

Update

Marked as "Fixed" by the client.
Addressed in: `7368bb88321376538bce979b5556977dc63ed303`.
The client provided the following explanation:

```
3. Ack - used in off chain components - won't fix
```

Description:

1. Code related to updating the minter of the UStb contract is defined but not used:
   - events `MinterAdded` and `MinterRemoved` in `IUStbDefinitions.sol`;
   - function `setMinter()` in `IUStb.sol`;
2. In `UStbMinting.sol`, the variable `EIP712_DOMAIN_TYPEHASH` is defined but not used;
3. In `UStbMinting.sol`, the function `verifyOrder()` returns the value `bytes32 taker_order_hash`. However, this value is not used by the functions calling `verifyOrder()`, which are `mint()` and `redeem()`. If not used, returning it might be unnecessary;
4. In `IUStbDefinitions.sol`, the event `ToggleTransfers` is not used;

Recommendation: Consider addressing the items listed above.

S3 Unchecked Blocks in `for` Loop

Acknowledged

Update

Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
won't fix
```

File(s) affected: `UStbMinting.sol`

Description: The code in scope uses version 0.8.26 of Solidity. Since the version 0.8.22 of Solidity, it no longer saves gas to wrap the counter increment of a `for` loop in an `unchecked` block. However, `unchecked` blocks are still being used for this purpose at the following locations:

1. `UStbMinting.sol#L178-180`;
2. `UStbMinting.sol#L192-194`;
3. `UStbMinting.sol#L504-506`;
4. `UStbMinting.sol#L591-593`;

More details.

Recommendation: Remove the `unchecked` blocks.

S4 Contract `UStb` Could Inherit the Interface `IUStb`

Acknowledged

Update
Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
won't fix
```

File(s) affected: `UStb.sol, IUStb.sol`

Description: The `UStb` contract currently does not inherit the `IUStb` interface that defines its `external` and `public` functions. Inheriting the interface would enforce its implementation, preventing any mistakes from arising in the future like missing functions in the case of further revision to the contract.

Recommendation: Consider `UStb` inheriting the interface `IUStb`.

S5 Internal Functions Used Once

Acknowledged

Update
Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
won't fix
```

File(s) affected: `UStbMinting.sol`

Description: Internal functions provide a convenient way to reduce code duplication across a codebase and bring down the overall size of smart contracts. However, when an internal function is used only once, it can increase a contract's code size and decrease readability. Examples where this occurs are the following:

1. `_setMaxRedeemPerBlock()`;
2. `_setMaxMintPerBlock()`;
3. `_transferToBeneficiary()`;
4. `_transferCollateral()`;

Recommendation: Remove the above internal functions and include their logic inside the calling functions.

S6 Redundant Check of the Blacklisted Role of `to` in Function `redistributeLockedAmount()`

Acknowledged

Update
Marked as "Acknowledged" by the client.
The client provided the following explanation:

```
won't fix
```

File(s) affected: `UStb.sol`

Description: In the function `redistributeLockedAmount()`, one of the necessary conditions for not reverting is `!hasRole(BLACKLISTED_ROLE, to)`, which necessitates that the `to` address is not blacklisted. However, this check is redundant, because the function `_mint()` at `UStb.sol#L101` will call the function `_beforeTokenTransfer()`, which stipulates that for every `TransferState`, the call must revert when `to` is blacklisted.

Recommendation: Remove the unnecessary check regarding whether `to` is blacklisted.

S7 Private Visibility of `_pendingDefaultAdmin` Makes the Monitoring of Critical Operations More Difficult

Acknowledged

Update
Marked as "Acknowledged" by the client.
The client provided the following explanation:

> `won't fix`

File(s) affected: `SingleAdminAccessControl.sol, SingleAdminAccessControlUpgradeable.sol`

Description: The fact that the storage variable `_pendingDefaultAdmin` is `private` in `SingleAdminAccessControl.sol` makes it harder for off-chain observers to monitor critical updates of the system. Only an agent monitoring the events `AdminTransferRequested` can observe that such a transfer of roles is ongoing. Note that the value of `_currentDefaultAdmin` can easily be accessed via the view function `owner()`.

Recommendation: Consider creating a view function or using the keyword `public` for `_pendingDefaultAdmin`.

Definitions

- **High severity** – High-severity issues usually put a large number of users' sensitive information at risk, or are reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
- **Medium severity** – Medium-severity issues tend to put a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or are reasonably likely to lead to moderate financial impact.
- **Low severity** – The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.
- **Informational** – The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
- **Undetermined** – The impact of the issue is uncertain.
- **Fixed** – Adjusted program implementation, requirements or constraints to eliminate the risk.
- **Mitigated** – Implemented actions to minimize the impact or likelihood of the risk.
- **Acknowledged** – The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Files

- `c5e...ba7 ./contracts/SingleAdminAccessControlUpgradeable.sol`
- `bbb...ed1 ./contracts/SingleAdminAccessControl.sol`
- `3d4...882 ./contracts/interfaces/ISingleAdminAccessControl.sol`
- `c46...72a ./contracts/ustb/IUStb.sol`
- `11a...59a ./contracts/ustb/UStbMinting.sol`
- `eb6...c2f ./contracts/ustb/IUStbMintingEvents.sol`
- `3bc...140 ./contracts/ustb/IUStbDefinitions.sol`
- `4c1...552 ./contracts/ustb/UStb.sol`
- `1e9...6ef ./contracts/ustb/IUStbMinting.sol`

Automated Analysis

N/A

Test Suite Results

The test suite has 125 tests, and all passed. It covers happy and unhappy paths. It also includes fuzzing tests.

** Fix review update **

The test suite has now 235 tests, and all passed. It covers happy and unhappy paths. It also includes fuzzing tests.

```
Ran 4 tests for test/foundry/test/UStbMinting.Whitelist.t.sol:UStbMintingWhitelistTest
[PASS] test_non_whitelisted_beneficiary_mint() (gas: 350979)
[PASS] test_non_whitelisted_beneficiary_redeem() (gas: 527048)
[PASS] test_whitelist_mint() (gas: 327420)
[PASS] test_whitelist_redeem() (gas: 372159)
Suite result: ok. 4 passed; 0 failed; 0 skipped; finished in 4.82s (15.86ms CPU time)
Ran 1 test for test/foundry/test/UStbMinting.SmartContractSigning.t.sol:UStbMintingContractSigningTest
[PASS] test_multi_sig_eip_1271_mint() (gas: 403916)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 4.82s (12.46ms CPU time)
Ran 6 tests for test/foundry/test/UStbMinting.StableRatios.t.sol:UStbMintingStableRatiosTest
[PASS] test_stable_ratios_minting_invalid() (gas: 357657)
[PASS] test_stable_ratios_redeem_invalid() (gas: 354893)
[PASS] test_stable_ratios_setup() (gas: 19746)
[PASS] test_stables_limit_minting_valid() (gas: 361442)
[PASS] test_stables_limit_redeem_valid() (gas: 524921)
[PASS] test_verify_stables_limit() (gas: 75243)
Suite result: ok. 6 passed; 0 failed; 0 skipped; finished in 4.91s (4.89ms CPU time)
Ran 21 tests for test/foundry/UStb.allTests.t.sol:UStbTest
[PASS] testAdminCanGrantRevokeBlacklistRole() (gas: 81003)
[PASS] testAdminCanGrantRevokeWhitelistRole() (gas: 105198)
[PASS] testBlacklistManagerCanGrantRevokeBlacklistRole() (gas: 109329)
[PASS] testInvalidMinter() (gas: 51975)
[PASS] testRandomAddressGrantRevokeBlackistWhitelistRoleException() (gas: 317626)
[PASS] testRedistributeLockedAmountBlacklistedToFails() (gas: 68509)
[PASS] testRedistributeLockedAmountNonAdmin() (gas: 88164)
[PASS] testRedistributeLockedAmountNotBlacklistedFromFails() (gas: 39898)
[PASS] testRedistributeLockedAmountPass() (gas: 85088)
[PASS] testRenounceRoleExpectRevert() (gas: 22589)
[PASS] testRescueTokenAdmin() (gas: 68570)
[PASS] testRescueTokenNonAdmin() (gas: 101580)
[PASS] testTransferStateFullyDisabled() (gas: 32162)
[PASS] testTransferStateFullyEnabledBlacklistedFromExpectRevert() (gas: 61297)
[PASS] testTransferStateFullyEnabledBlacklistedToExpectRevert() (gas: 61242)
[PASS] testTransferStateWhitelistEnabledFail() (gas: 39065)
[PASS] testTransferStateWhitelistEnabledFail2() (gas: 67021)
[PASS] testTransferStateWhitelistEnabledFail3() (gas: 147084)
[PASS] testTransferStateWhitelistEnabledFail4() (gas: 125645)
[PASS] testTransferStateWhitelistEnabledPass() (gas: 112516)
[PASS] testWhitelistManagerCanGrantRevokeWhitelistRole() (gas: 127051)
Suite result: ok. 21 passed; 0 failed; 0 skipped; finished in 4.93s (6.12ms CPU time)
Ran 5 tests for test/foundry/test/UStbMinting.Delegate.t.sol:UStbMintingDelegateTest
[PASS] testCanUndelegate() (gas: 170389)
[PASS] testDelegateFailureMint() (gas: 155198)
[PASS] testDelegateFailureRedeem() (gas: 368601)
[PASS] testDelegateSuccessfulMint() (gas: 325972)
[PASS] testDelegateSuccessfulRedeem() (gas: 387762)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 5.02s (5.37ms CPU time)
Ran 19 tests for test/foundry/test/UStbMinting.core.t.sol:UStbMintingCoreTest
[PASS] test_add_and_remove_supported_asset() (gas: 56220)
[PASS] test_cannotAdd_UStb_revert() (gas: 27120)
[PASS] test_cannotAdd_addressZero_revert() (gas: 22851)
[PASS] test_cannot_add_asset_already_supported_revert() (gas: 74554)
[PASS] test_cannot_removeAsset_not_supported_revert() (gas: 19345)
[PASS] test_expired_orders_revert() (gas: 128974)
[PASS] test_fuzz_mint_noSlippage(uint128) (runs: 1000, μ: 276460, ~: 276460)
[PASS] test_fuzz_multipleInvalid_custodyRatios_revert(uint128) (runs: 1000, μ: 145971, ~: 146011)
[PASS] test_fuzz_singleInvalid_custodyRatio_revert(uint128) (runs: 1000, μ: 133683, ~: 133696)
[PASS] test_mint() (gas: 262579)
[PASS] test_multipleValid_custodyRatios_addresses() (gas: 432529)
[PASS] test_nativeEth_withdraw() (gas: 365797)
[PASS] test_receive_eth() (gas: 20154)
[PASS] test_redeem() (gas: 343903)
[PASS] test_redeem_invalidNonce_revert() (gas: 366926)
[PASS] test_sending_mint_order_to_redeem_revert() (gas: 109002)
[PASS] test_sending_redeem_order_to_mint_revert() (gas: 328025)
[PASS] test_unsupported_assets_ERC20_revert() (gas: 101453)
[PASS] test_unsupported_assets_ETH_revert() (gas: 181156)
Suite result: ok. 19 passed; 0 failed; 0 skipped; finished in 5.86s (2.36s CPU time)
```

```
Ran 10 tests for test/foundry/test/UStbMinting.blockLimits.t.sol:UStbMintingBlockLimitsTest
[PASS] test_fuzz_maxMint_perBlock_exceeded_revert(uint128) (runs: 1000, μ: 120830, ~: 120830)
[PASS] test_fuzz_maxMint_perBlock_setter(uint128) (runs: 1000, μ: 33787, ~: 33787)
[PASS] test_fuzz_maxRedeem_perBlock_exceeded_revert(uint128) (runs: 1000, μ: 355918, ~: 355918)
[PASS] test_fuzz_maxRedeem_perBlock_setter(uint128) (runs: 1000, μ: 33755, ~: 33755)
[PASS] test_fuzz_mint_maxMint_perBlock_exceeded_revert(uint128) (runs: 1000, μ: 120600, ~: 120600)
[PASS] test_fuzz_nextBlock_mint_is_zero(uint128) (runs: 1000, μ: 272230, ~: 272230)
[PASS] test_fuzz_nextBlock_redeem_is_zero(uint128) (runs: 1000, μ: 344934, ~: 344939)
[PASS] test_global_mint_limit_versus_local_perBlock() (gas: 348499)
[PASS] test_multiple_mints() (gas: 338984)
[PASS] test_multiple_redeem() (gas: 530729)
Suite result: ok. 10 passed; 0 failed; 0 skipped; finished in 6.87s (5.12s CPU time)
Ran 59 tests for test/foundry/test/UStbMinting.ACL.t.sol:UStbMintingACLTest
[PASS] testAdminCanCancelTransfer() (gas: 41770)
[PASS] testCanRepeatedlyTransferAdmin() (gas: 46481)
[PASS] testCanTransferOwnership() (gas: 62109)
[PASS] testCancelTransferAdmin() (gas: 40428)
[PASS] testCorrectInitConfig() (gas: 4979615)
[PASS] testInitConfigBlockLimitMismatch() (gas: 5311205)
[PASS] testNewOwnerCanPerformOwnerActions() (gas: 88626)
[PASS] testNonAdminCanRenounceRoles() (gas: 35638)
[PASS] testOldOwnerCantPerformOwnerActions() (gas: 95462)
[PASS] testOldOwnerCantTransferOwnership() (gas: 93294)
[PASS] testOwnershipCannotBeRenounced() (gas: 24796)
[PASS] testOwnershipTransferRequiresTwoSteps() (gas: 51527)
[PASS] test_admin_can_add_gatekeeper() (gas: 44480)
[PASS] test_admin_can_add_minter() (gas: 44384)
[PASS] test_admin_can_disable_mint(bool) (runs: 1000, μ: 78663, ~: 129614)
[PASS] test_admin_can_disable_redeem(bool) (runs: 1000, μ: 189886, ~: 346658)
[PASS] test_admin_can_enable_mint() (gas: 281917)
[PASS] test_admin_can_enable_redeem() (gas: 353196)
[PASS] test_admin_can_remove_gatekeeper() (gas: 37092)
[PASS] test_admin_can_remove_minter() (gas: 37048)
[PASS] test_admin_cannot_transfer_self() (gas: 21968)
[PASS] test_base_transferAdmin() (gas: 65049)
[PASS] test_collateralManager_canTransferNative_custody() (gas: 143305)
[PASS] test_collateralManager_canTransfer_custody() (gas: 151908)
[PASS] test_collateralManager_cannotTransfer_zeroAddress() (gas: 140616)
[PASS] test_fuzz_nonAdmin_cannot_enable_mint_revert(address) (runs: 1000, μ: 180622, ~: 180622)
[PASS] test_fuzz_nonAdmin_cannot_enable_redeem_revert(address) (runs: 1000, μ: 397708, ~: 397708)
[PASS] test_fuzz_nonCollateralManager_cannot_transferCustody_revert(address) (runs: 1000, μ: 99602, ~:
99602)
[PASS] test_fuzz_nonOwner_cannot_add_supportedAsset_revert(address) (runs: 1000, μ: 47388, ~: 47388)
[PASS] test_fuzz_nonOwner_cannot_remove_supportedAsset_revert(address) (runs: 1000, μ: 100553, ~: 100553)
[PASS] test_fuzz_notAdmin_cannot_add_gatekeeper(address) (runs: 1000, μ: 64182, ~: 64182)
[PASS] test_fuzz_notAdmin_cannot_add_minter(address) (runs: 1000, μ: 64116, ~: 64116)
[PASS] test_fuzz_notAdmin_cannot_remove_gatekeeper(address) (runs: 1000, μ: 93412, ~: 93412)
[PASS] test_fuzz_notAdmin_cannot_remove_minter(address) (runs: 1000, μ: 93290, ~: 93290)
[PASS] test_fuzz_notMinter_cannot_mint(address) (runs: 1000, μ: 145339, ~: 145339)
[PASS] test_fuzz_not_gatekeeper_cannot_disable_mintRedeem_revert(address) (runs: 1000, μ: 62502, ~:
62502)
[PASS] test_fuzz_not_gatekeeper_cannot_remove_collateral_manager_revert(address) (runs: 1000, μ: 63199,
~: 63199)
[PASS] test_fuzz_not_gatekeeper_cannot_remove_minter_revert(address) (runs: 1000, μ: 63221, ~: 63221)
[PASS] test_fuzz_not_gatekeeper_cannot_remove_redeemer_revert(address) (runs: 1000, μ: 63177, ~: 63177)
[PASS] test_gatekeeper_can_disable_mintRedeem() (gas: 146289)
[PASS] test_gatekeeper_can_remove_collateral_manager() (gas: 21435)
[PASS] test_gatekeeper_can_remove_minter() (gas: 21457)
[PASS] test_gatekeeper_can_remove_redeemer() (gas: 21411)
[PASS] test_gatekeeper_cannot_add_collateral_managers_revert() (gas: 63408)
[PASS] test_gatekeeper_cannot_add_minters_revert() (gas: 63330)
[PASS] test_gatekeeper_cannot_enable_mint_revert() (gas: 182620)
[PASS] test_gatekeeper_cannot_enable_redeem_revert() (gas: 399663)
[PASS] test_grantRole_AdminRoleExternally() (gas: 43030)
[PASS] test_grantRole_nonAdminRole() (gas: 43641)
[PASS] test_redeem_notRedeemer_revert() (gas: 368145)
[PASS] test_renounceRole_AdminRole() (gas: 15656)
[PASS] test_renounceRole_forDifferentAccount() (gas: 15448)
[PASS] test_renounceRole_nonAdminRole() (gas: 34142)
[PASS] test_renounceRole_notAdmin() (gas: 17624)
[PASS] test_revokeRole_AdminRole() (gas: 15614)
[PASS] test_revokeRole_nonAdminRole() (gas: 34263)
```

```
[PASS] test_revokeRole_notAdmin() (gas: 44703)
[PASS] test_revoke_AdminRole() (gas: 17929)
[PASS] test_transferAdmin_notAdmin() (gas: 40444)
Suite result: ok. 59 passed; 0 failed; 0 skipped; finished in 6.87s (7.04s CPU time)
Ran 8 test suites in 6.88s (44.11s CPU time): 125 tests passed, 0 failed, 0 skipped (125 total tests)

** Fix review update **

Ran 7 tests for test/foundry/UStb.admin.t.sol:UStbTest
[PASS] testAdminCanGrantRevokeBlacklistRole() (gas: 109257)
[PASS] testAdminCanGrantRevokeWhitelistRole() (gas: 127873)
[PASS] testBlacklistManagerCanGrantRevokeBlacklistRole() (gas: 139135)
[PASS] testInvalidMinter() (gas: 62262)
[PASS] testRandomAddressGrantRevokeBlackistWhitelistRoleException() (gas: 397015)
[PASS] testRenounceRoleExpectRevert() (gas: 23775)
[PASS] testWhitelistManagerCanGrantRevokeWhitelistRole() (gas: 153585)
Suite result: ok. 7 passed; 0 failed; 0 skipped; finished in 3.67s (25.83ms CPU time)
Ran 123 tests for test/foundry/UStb.transfers.t.sol:UStbTransferTest
[PASS] testRedistributeLockedAmountBlacklistedToFails() (gas: 73156)
[PASS] testRedistributeLockedAmountNonAdmin() (gas: 99881)
[PASS] testRedistributeLockedAmountNotBlacklistedFromFails() (gas: 43375)
[PASS] testRedistributeLockedAmountPass() (gas: 94956)
[PASS] testRedistributeLockedAmountWhitelistEnabledPass() (gas: 133518)
[PASS] testRescueTokenAdmin() (gas: 88664)
[PASS] testRescueTokenNonAdmin() (gas: 121447)
[PASS] testTransferStateFullyDisabled() (gas: 35289)
[PASS] testTransferStateFullyEnabledBlacklistedFromExpectRevert() (gas: 65713)
[PASS] testTransferStateFullyEnabledBlacklistedToExpectRevert() (gas: 65692)
[PASS] testTransferStateWhitelistEnabledFail() (gas: 48313)
[PASS] testTransferStateWhitelistEnabledFail2() (gas: 78190)
[PASS] testTransferStateWhitelistEnabledFail3() (gas: 161881)
[PASS] testTransferStateWhitelistEnabledFail4() (gas: 139004)
[PASS] testTransferStateWhitelistEnabledPass() (gas: 126932)
[PASS] test_bl_sender_bl_from_bl_to_fully_disabled_revert() (gas: 93202)
[PASS] test_bl_sender_bl_from_bl_to_fully_enabled_revert() (gas: 95953)
[PASS] test_bl_sender_bl_from_bl_to_whitelist_enabled_revert() (gas: 106291)
[PASS] test_bl_sender_bl_from_burn_fully_disabled_revert() (gas: 65401)
[PASS] test_bl_sender_bl_from_burn_fully_enabled_revert() (gas: 64512)
[PASS] test_bl_sender_bl_from_burn_whitelist_enabled_revert() (gas: 78403)
[PASS] test_bl_sender_bl_from_to_fully_disabled_revert() (gas: 65359)
[PASS] test_bl_sender_bl_from_to_fully_enabled_revert() (gas: 68150)
[PASS] test_bl_sender_bl_from_to_whitelist_enabled_revert() (gas: 78469)
[PASS] test_bl_sender_bl_from_wl_to_fully_disabled_revert() (gas: 95158)
[PASS] test_bl_sender_bl_from_wl_to_fully_enabled_revert() (gas: 97977)
[PASS] test_bl_sender_bl_from_wl_to_whitelist_enabled_revert() (gas: 108225)
[PASS] test_bl_sender_from_bl_to_fully_disabled_revert() (gas: 126367)
[PASS] test_bl_sender_from_bl_to_fully_enabled_revert() (gas: 129115)
[PASS] test_bl_sender_from_bl_to_whitelist_enabled_revert() (gas: 139391)
[PASS] test_bl_sender_from_burn_fully_disabled_revert() (gas: 94905)
[PASS] test_bl_sender_from_burn_fully_enabled_revert() (gas: 97408)
[PASS] test_bl_sender_from_burn_whitelist_enabled_revert() (gas: 107993)
[PASS] test_bl_sender_from_to_fully_disabled_revert() (gas: 98521)
[PASS] test_bl_sender_from_to_fully_enabled_revert() (gas: 101338)
[PASS] test_bl_sender_from_to_whitelist_enabled_revert() (gas: 111570)
[PASS] test_bl_sender_from_wl_to_fully_disabled_revert() (gas: 128321)
[PASS] test_bl_sender_from_wl_to_fully_enabled_revert() (gas: 131096)
[PASS] test_bl_sender_from_wl_to_whitelist_enabled_revert() (gas: 141434)
[PASS] test_bl_sender_wl_from_bl_to_fully_disabled_revert() (gas: 156120)
[PASS] test_bl_sender_wl_from_bl_to_fully_enabled_revert() (gas: 158918)
[PASS] test_bl_sender_wl_from_bl_to_whitelist_enabled_revert() (gas: 169212)
[PASS] test_bl_sender_wl_from_burn_fully_disabled_revert() (gas: 124704)
[PASS] test_bl_sender_wl_from_burn_fully_enabled_revert() (gas: 127523)
[PASS] test_bl_sender_wl_from_burn_whitelist_enabled_revert() (gas: 137818)
[PASS] test_bl_sender_wl_from_to_fully_disabled_revert() (gas: 128342)
[PASS] test_bl_sender_wl_from_to_fully_enabled_revert() (gas: 131161)
[PASS] test_bl_sender_wl_from_to_whitelist_enabled_revert() (gas: 141347)
[PASS] test_bl_sender_wl_from_wl_to_fully_disabled_revert() (gas: 156100)
[PASS] test_bl_sender_wl_from_wl_to_fully_enabled_revert() (gas: 158959)
[PASS] test_bl_sender_wl_from_wl_to_whitelist_enabled_revert() (gas: 169191)
[PASS] test_sender_bl_from_bl_to_fully_disabled_revert() (gas: 126364)
[PASS] test_sender_bl_from_bl_to_fully_enabled_revert() (gas: 131520)
[PASS] test_sender_bl_from_bl_to_whitelist_enabled_revert() (gas: 139346)
```

```
[PASS] test_sender_bl_from_burn_fully_disabled_revert() (gas: 94948)
[PASS] test_sender_bl_from_burn_fully_enabled_revert() (gas: 100057)
[PASS] test_sender_bl_from_burn_whitelist_enabled_revert() (gas: 108019)
[PASS] test_sender_bl_from_to_fully_disabled_revert() (gas: 98522)
[PASS] test_sender_bl_from_to_fully_enabled_revert() (gas: 103740)
[PASS] test_sender_bl_from_to_whitelist_enabled_revert() (gas: 111568)
[PASS] test_sender_bl_from_wl_to_fully_disabled_revert() (gas: 128365)
[PASS] test_sender_bl_from_wl_to_fully_enabled_revert() (gas: 133453)
[PASS] test_sender_bl_from_wl_to_whitelist_enabled_revert() (gas: 141368)
[PASS] test_sender_from_bl_to_fully_disabled_revert() (gas: 65379)
[PASS] test_sender_from_bl_to_fully_enabled_revert() (gas: 70622)
[PASS] test_sender_from_bl_to_whitelist_enabled_revert() (gas: 78470)
[PASS] test_sender_from_burn_fully_disabled_revert() (gas: 31870)
[PASS] test_sender_from_burn_fully_enabled_success() (gas: 51903)
[PASS] test_sender_from_burn_whitelist_enabled_revert() (gas: 44962)
[PASS] test_sender_from_to_fully_disabled_revert() (gas: 35222)
[PASS] test_sender_from_to_fully_enabled_success() (gas: 54711)
[PASS] test_sender_from_to_whitelist_enabled_revert() (gas: 48669)
[PASS] test_sender_from_wl_to_fully_disabled_revert() (gas: 65379)
[PASS] test_sender_from_wl_to_fully_enabled_success() (gas: 89496)
[PASS] test_sender_from_wl_to_whitelist_enabled_revert() (gas: 78427)
[PASS] test_sender_wl_from_bl_to_fully_disabled_revert() (gas: 128322)
[PASS] test_sender_wl_from_bl_to_fully_enabled_revert() (gas: 135881)
[PASS] test_sender_wl_from_bl_to_whitelist_enabled_revert() (gas: 141390)
[PASS] test_sender_wl_from_burn_fully_disabled_revert() (gas: 94926)
[PASS] test_sender_wl_from_burn_fully_enabled_success() (gas: 102084)
[PASS] test_sender_wl_from_burn_whitelist_enabled_reverts() (gas: 107974)
[PASS] test_sender_wl_from_to_fully_disabled_revert() (gas: 98522)
[PASS] test_sender_wl_from_to_fully_enabled_success() (gas: 104673)
[PASS] test_sender_wl_from_to_whitelist_enabled_revert() (gas: 111613)
[PASS] test_sender_wl_from_wl_to_fully_disabled_revert() (gas: 126298)
[PASS] test_sender_wl_from_wl_to_fully_enabled_success() (gas: 132472)
[PASS] test_sender_wl_from_wl_to_whitelist_enabled_revert() (gas: 139346)
[PASS] test_wl_sender_bl_from_bl_to_fully_disabled_revert() (gas: 154841)
[PASS] test_wl_sender_bl_from_bl_to_fully_enabled_revert() (gas: 161252)
[PASS] test_wl_sender_bl_from_bl_to_whitelist_enabled_revert() (gas: 168307)
[PASS] test_wl_sender_bl_from_burn_fully_disabled_revert() (gas: 128366)
[PASS] test_wl_sender_bl_from_burn_fully_enabled_revert() (gas: 133454)
[PASS] test_wl_sender_bl_from_burn_whitelist_enabled_revert() (gas: 142058)
[PASS] test_wl_sender_bl_from_to_fully_disabled_revert() (gas: 124770)
[PASS] test_wl_sender_bl_from_to_fully_enabled_revert() (gas: 133477)
[PASS] test_wl_sender_bl_from_to_whitelist_enabled_revert() (gas: 138529)
[PASS] test_wl_sender_bl_from_wl_to_fully_disabled_revert() (gas: 156167)
[PASS] test_wl_sender_bl_from_wl_to_fully_enabled_revert() (gas: 161276)
[PASS] test_wl_sender_bl_from_wl_to_whitelist_enabled_revert() (gas: 169904)
[PASS] test_wl_sender_from_bl_to_fully_disabled_revert() (gas: 128365)
[PASS] test_wl_sender_from_bl_to_fully_enabled_revert() (gas: 135878)
[PASS] test_wl_sender_from_bl_to_whitelist_enabled_revert() (gas: 142104)
[PASS] test_wl_sender_from_burn_fully_disabled_revert() (gas: 124725)
[PASS] test_wl_sender_from_burn_fully_enabled_success() (gas: 101747)
[PASS] test_wl_sender_from_burn_whitelist_enabled_revert() (gas: 108686)
[PASS] test_wl_sender_from_to_fully_disabled_revert() (gas: 98522)
[PASS] test_wl_sender_from_to_fully_enabled_success() (gas: 105010)
[PASS] test_wl_sender_from_to_whitelist_enabled_revert() (gas: 112303)
[PASS] test_wl_sender_from_wl_to_fully_disabled_revert() (gas: 126322)
[PASS] test_wl_sender_from_wl_to_fully_enabled_success() (gas: 132851)
[PASS] test_wl_sender_from_wl_to_whitelist_enabled_revert() (gas: 140058)
[PASS] test_wl_sender_wl_from_bl_to_fully_disabled_revert() (gas: 95159)
[PASS] test_wl_sender_wl_from_bl_to_fully_enabled_revert() (gas: 100714)
[PASS] test_wl_sender_wl_from_bl_to_whitelist_enabled_revert() (gas: 109339)
[PASS] test_wl_sender_wl_from_burn_fully_disabled_revert() (gas: 61674)
[PASS] test_wl_sender_wl_from_burn_fully_enabled_success() (gas: 86416)
[PASS] test_wl_sender_wl_from_burn_whitelist_enabled_success() (gas: 89796)
[PASS] test_wl_sender_wl_from_to_fully_disabled_revert() (gas: 65445)
[PASS] test_wl_sender_wl_from_to_fully_enabled_success() (gas: 89832)
[PASS] test_wl_sender_wl_from_to_whitelist_enabled_revert() (gas: 79540)
[PASS] test_wl_sender_wl_from_wl_to_fully_disabled_revert() (gas: 93203)
[PASS] test_wl_sender_wl_from_wl_to_fully_enabled_success() (gas: 117610)
[PASS] test_wl_sender_wl_from_wl_to_whitelist_enabled_success() (gas: 127311)
Suite result: ok. 123 passed; 0 failed; 0 skipped; finished in 3.78s (189.03ms CPU time)
Ran 6 tests for test/foundry/test/UStbMinting.StableRatios.t.sol:UStbMintingStableRatiosTest
[PASS] test_stable_ratios_minting_invalid() (gas: 412498)
```

```
[PASS] test_stable_ratios_redeem_invalid() (gas: 409339)
[PASS] test_stable_ratios_setup() (gas: 20976)
[PASS] test_stables_limit_minting_valid() (gas: 427886)
[PASS] test_stables_limit_redeem_valid() (gas: 653919)
[PASS] test_verify_stables_limit() (gas: 82668)
Suite result: ok. 6 passed; 0 failed; 0 skipped; finished in 5.40s (63.39ms CPU time)
Ran 1 test for test/foundry/test/UStbMinting.SmartContractSigning.t.sol:UStbMintingContractSigningTest
[PASS] test_multi_sig_eip_1271_mint() (gas: 452300)
Suite result: ok. 1 passed; 0 failed; 0 skipped; finished in 5.81s (8.05ms CPU time)
Ran 19 tests for test/foundry/test/UStbMinting.core.t.sol:UStbMintingCoreTest
[PASS] test_add_and_remove_supported_asset() (gas: 61247)
[PASS] test_cannotAdd_UStb_revert() (gas: 29397)
[PASS] test_cannotAdd_addressZero_revert() (gas: 25082)
[PASS] test_cannot_add_asset_already_supported_revert() (gas: 80390)
[PASS] test_cannot_removeAsset_not_supported_revert() (gas: 21148)
[PASS] test_expired_orders_revert() (gas: 154255)
[PASS] test_fuzz_mint_noSlippage(uint128) (runs: 1000, μ: 316580, ~: 316580)
[PASS] test_fuzz_multipleInvalid_custodyRatios_revert(uint128) (runs: 1000, μ: 174760, ~: 174805)
[PASS] test_fuzz_singleInvalid_custodyRatio_revert(uint128) (runs: 1000, μ: 160523, ~: 160547)
[PASS] test_mint() (gas: 294303)
[PASS] test_multipleValid_custodyRatios_addresses() (gas: 497811)
[PASS] test_nativeEth_withdraw() (gas: 430249)
[PASS] test_receive_eth() (gas: 21860)
[PASS] test_redeem() (gas: 394408)
[PASS] test_redeem_invalidNonce_revert() (gas: 423112)
[PASS] test_sending_mint_order_to_redeem_revert() (gas: 128698)
[PASS] test_sending_redeem_order_to_mint_revert() (gas: 378353)
[PASS] test_unsupported_assets_ERC20_revert() (gas: 121165)
[PASS] test_unsupported_assets_ETH_revert() (gas: 206013)
Suite result: ok. 19 passed; 0 failed; 0 skipped; finished in 12.99s (16.50s CPU time)
Ran 5 tests for test/foundry/test/UStbMinting.Delegate.t.sol:UStbMintingDelegateTest
[PASS] testCanUndelegate() (gas: 211401)
[PASS] testDelegateFailureMint() (gas: 191907)
[PASS] testDelegateFailureRedeem() (gas: 433616)
[PASS] testDelegateSuccessfulMint() (gas: 373012)
[PASS] testDelegateSuccessfulRedeem() (gas: 455167)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 17.93s (39.64ms CPU time)
Ran 10 tests for test/foundry/test/UStbMinting.blockLimits.t.sol:UStbMintingBlockLimitsTest
[PASS] test_fuzz_maxMint_perBlock_exceeded_revert(uint128) (runs: 1000, μ: 146333, ~: 146333)
[PASS] test_fuzz_maxMint_perBlock_setter(uint128) (runs: 1000, μ: 38137, ~: 38137)
[PASS] test_fuzz_maxRedeem_perBlock_exceeded_revert(uint128) (runs: 1000, μ: 411857, ~: 411860)
[PASS] test_fuzz_maxRedeem_perBlock_setter(uint128) (runs: 1000, μ: 38035, ~: 38035)
[PASS] test_fuzz_mint_maxMint_perBlock_exceeded_revert(uint128) (runs: 1000, μ: 145509, ~: 145509)
[PASS] test_fuzz_nextBlock_mint_is_zero(uint128) (runs: 1000, μ: 306064, ~: 306064)
[PASS] test_fuzz_nextBlock_redeem_is_zero(uint128) (runs: 1000, μ: 394328, ~: 394332)
[PASS] test_global_mint_limit_versus_local_perBlock() (gas: 395191)
[PASS] test_multiple_mints() (gas: 404563)
[PASS] test_multiple_redeem() (gas: 629008)
Suite result: ok. 10 passed; 0 failed; 0 skipped; finished in 17.93s (36.13s CPU time)
Ran 5 tests for test/foundry/test/UStbMinting.Whitelist.t.sol:UStbMintingWhitelistTest
[PASS] test_non_whitelisted_beneficiary_mint() (gas: 391495)
[PASS] test_non_whitelisted_beneficiary_redeem() (gas: 597978)
[PASS] test_whitelist_mint() (gas: 368278)
[PASS] test_whitelist_redeem() (gas: 430404)
[PASS] test_whitelisted_beneficiary_whitelist_enabled_transfer_redeem() (gas: 638514)
Suite result: ok. 5 passed; 0 failed; 0 skipped; finished in 19.50s (60.11ms CPU time)
Ran 59 tests for test/foundry/test/UStbMinting.ACL.t.sol:UStbMintingACLTest
[PASS] testAdminCanCancelTransfer() (gas: 45619)
[PASS] testCanRepeatedlyTransferAdmin() (gas: 47837)
[PASS] testCanTransferOwnership() (gas: 64980)
[PASS] testCancelTransferAdmin() (gas: 44033)
[PASS] testCorrectInitConfig() (gas: 6976382)
[PASS] testInitConfigBlockLimitMismatch() (gas: 7329714)
[PASS] testNewOwnerCanPerformOwnerActions() (gas: 93117)
[PASS] testNonAdminCanRenounceRoles() (gas: 39085)
[PASS] testOldOwnerCantPerformOwnerActions() (gas: 110267)
[PASS] testOldOwnerCantTransferOwnership() (gas: 107968)
[PASS] testOwnershipCannotBeRenounced() (gas: 28396)
[PASS] testOwnershipTransferRequiresTwoSteps() (gas: 55679)
[PASS] test_admin_can_add_gatekeeper() (gas: 46802)
[PASS] test_admin_can_add_minter() (gas: 46653)
[PASS] test_admin_can_disable_mint(bool) (runs: 1000, μ: 92841, ~: 30689)
```

```
[PASS] test_admin_can_disable_redeem(bool) (runs: 1000, μ: 209875, ~: 25763)
[PASS] test_admin_can_enable_mint() (gas: 320525)
[PASS] test_admin_can_enable_redeem() (gas: 406411)
[PASS] test_admin_can_remove_gatekeeper() (gas: 40868)
[PASS] test_admin_can_remove_minter() (gas: 40818)
[PASS] test_admin_cannot_transfer_self() (gas: 23936)
[PASS] test_base_transferAdmin() (gas: 69589)
[PASS] test_collateralManager_canTransferNative_custody() (gas: 148481)
[PASS] test_collateralManager_canTransfer_custody() (gas: 160244)
[PASS] test_collateralManager_cannotTransfer_zeroAddress() (gas: 144944)
[PASS] test_fuzz_nonAdmin_cannot_enable_mint_revert(address) (runs: 1000, μ: 225935, ~: 225935)
[PASS] test_fuzz_nonAdmin_cannot_enable_redeem_revert(address) (runs: 1000, μ: 472154, ~: 472154)
[PASS] test_fuzz_nonCollateralManager_cannot_transferCustody_revert(address) (runs: 1000, μ: 115818, ~:
115818)
[PASS] test_fuzz_nonOwner_cannot_add_supportedAsset_revert(address) (runs: 1000, μ: 59167, ~: 59167)
[PASS] test_fuzz_nonOwner_cannot_remove_supportedAsset_revert(address) (runs: 1000, μ: 114924, ~: 114924)
[PASS] test_fuzz_notAdmin_cannot_add_gatekeeper(address) (runs: 1000, μ: 80059, ~: 80059)
[PASS] test_fuzz_notAdmin_cannot_add_minter(address) (runs: 1000, μ: 79973, ~: 79973)
[PASS] test_fuzz_notAdmin_cannot_remove_gatekeeper(address) (runs: 1000, μ: 111678, ~: 111678)
[PASS] test_fuzz_notAdmin_cannot_remove_minter(address) (runs: 1000, μ: 111503, ~: 111503)
[PASS] test_fuzz_notMinter_cannot_mint(address) (runs: 1000, μ: 180528, ~: 180528)
[PASS] test_fuzz_not_gatekeeper_cannot_disable_mintRedeem_revert(address) (runs: 1000, μ: 77043, ~:
77043)
[PASS] test_fuzz_not_gatekeeper_cannot_remove_collateral_manager_revert(address) (runs: 1000, μ: 78780,
~: 78780)
[PASS] test_fuzz_not_gatekeeper_cannot_remove_minter_revert(address) (runs: 1000, μ: 78846, ~: 78846)
[PASS] test_fuzz_not_gatekeeper_cannot_remove_redeemer_revert(address) (runs: 1000, μ: 78757, ~: 78757)
[PASS] test_gatekeeper_can_disable_mintRedeem() (gas: 171572)
[PASS] test_gatekeeper_can_remove_collateral_manager() (gas: 23403)
[PASS] test_gatekeeper_can_remove_minter() (gas: 23469)
[PASS] test_gatekeeper_can_remove_redeemer() (gas: 23468)
[PASS] test_gatekeeper_cannot_add_collateral_managers_revert() (gas: 79022)
[PASS] test_gatekeeper_cannot_add_minters_revert() (gas: 78935)
[PASS] test_gatekeeper_cannot_enable_mint_revert() (gas: 227794)
[PASS] test_gatekeeper_cannot_enable_redeem_revert() (gas: 473925)
[PASS] test_grantRole_AdminRoleExternally() (gas: 53255)
[PASS] test_grantRole_nonAdminRole() (gas: 45838)
[PASS] test_redeem_notRedeemer_revert() (gas: 429960)
[PASS] test_renounceRole_AdminRole() (gas: 16649)
[PASS] test_renounceRole_forDifferentAccount() (gas: 16659)
[PASS] test_renounceRole_nonAdminRole() (gas: 36736)
[PASS] test_renounceRole_notAdmin() (gas: 18561)
[PASS] test_revokeRole_AdminRole() (gas: 16601)
[PASS] test_revokeRole_nonAdminRole() (gas: 36767)
[PASS] test_revokeRole_notAdmin() (gas: 54916)
[PASS] test_revoke_AdminRole() (gas: 19055)
[PASS] test_transferAdmin_notAdmin() (gas: 50191)
Suite result: ok. 59 passed; 0 failed; 0 skipped; finished in 19.50s (52.55s CPU time)
Ran 9 test suites in 19.51s (106.53s CPU time): 235 tests passed, 0 failed, 0 skipped (235 total tests)
```

Code Coverage

The coverage results for the relevant contracts could be improved. It is recommended for these contracts to reach scores above 90%.

** Fix review update **

The coverage results remain under 90% for the branches and could be improved.

File	% Lines	% Statements	% Branches	% Funcs
contracts/SingleAdminAccessControl.sol	100.00% (**16/**16)	94.74% (**18/**19)	75.00% (**3/**4)	100.00% (**8/**8)
contracts/SingleAdminAccessControlUpgradeable.sol	56.25% (**9/**16)	47.37% (**9/**19)	25.00% (**1/**4)	50.00% (**4/**8)

File	% Lines	% Statements	% Branches	% Funcs
contracts/lib/Upgrades.sol	15.56% (**7/**45)	20.31% (**13/**64)	100.00% (**0/**0)	14.71% (**5/**34)
contracts/lib/math/ABDKMathQuad.sol	0.00% (**0/**698)	0.00% (**0/**986)	0.00% (**0/**388)	0.00% (**0/**29)
contracts/lib/math/PRBMath.sol	0.00% (**0/**247)	0.00% (**0/**324)	0.00% (**0/**96)	0.00% (**0/**6)
contracts/lib/math/PRBMathSD59×18.sol	0.00% (**0/**183)	0.00% (**0/**362)	0.00% (**0/**57)	0.00% (**0/**22)
contracts/mock/MockMultisigWallet.sol	100.00% (**16/**16)	100.00% (**20/**20)	71.43% (**5/**7)	100.00% (**4/**4)
contracts/mock/MockToken.sol	100.00% (**6/**6)	100.00% (**6/**6)	50.00% (**1/**2)	100.00% (**4/**4)
contracts/mock/MockUSDT.sol	0.00% (**0/**3)	0.00% (**0/**3)	100.00% (**0/**0)	0.00% (**0/**3)
contracts/ustb/UStb.sol	97.62% (**41/**42)	97.06% (**66/**68)	66.67% (**8/**12)	100.00% (**12/**12)
contracts/ustb/UStbMinting.sol	90.86% (**179/**197)	87.18% (**238/**273)	63.49% (**40/**63)	81.25% (**39/**48)
script/DeploymentUtils.sol	0.00% (**0/**37)	0.00% (**0/**51)	0.00% (**0/**7)	0.00% (**0/**10)
script/ustb/DeployUStb.sol	0.00% (**0/**25)	0.00% (**0/**27)	0.00% (**0/**7)	0.00% (**0/**2)
script/ustb/DeployUStbMinting.sol	0.00% (**0/**23)	0.00% (**0/**30)	100.00% (**0/**0)	0.00% (**0/**2)
test/foundry/UStbBaseSetup.sol	100.00% (**43/**43)	100.00% (**44/**44)	100.00% (**1/**1)	100.00% (**1/**1)
test/foundry/UStbMinting.utils.sol	100.00% (**29/**29)	100.00% (**30/**30)	100.00% (**0/**0)	0.00% (**0/**4)
test/foundry/UStbMintingBaseSetup.sol	99.47% (**186/**187)	99.01% (**201/**203)	100.00% (**3/**3)	42.86% (**3/**7)
test/utils/SigUtils.sol	33.33% (**1/**3)	20.00% (**1/**5)	100.00% (**0/**0)	33.33% (**1/**3)
test/utils/Utils.sol	0.00% (**0/**11)	0.00% (**0/**16)	100.00% (**0/**0)	0.00% (**0/**3)
Total	29.17% (**533/**1827)	25.33% (**646/**2550)	9.52% (**62/**651)	38.57% (**81/**210)

** Fix review update **

File	% Lines	% Statements	% Branches	% Funcs
contracts/SingleAdminAccessControl.sol	100.00% (**16/**16)	94.74% (**18/**19)	75.00% (**3/**4)	100.00% (**8/**8)
contracts/SingleAdminAccessControlUpgradeable.sol	56.25% (**9/**16)	47.37% (**9/**19)	25.00% (**1/**4)	50.00% (**4/**8)
contracts/lib/Upgrades.sol	15.56% (**7/**45)	20.31% (**13/**64)	100.00% (**0/**0)	14.71% (**5/**34)

File	% Lines	% Statements	% Branches	% Funcs
contracts/lib/math/ABDKMathQuad.sol	0.00% (**0**/698)	0.00% (**0**/986)	0.00% (**0**/388)	0.00% (**0**/29)
contracts/lib/math/PRBMath.sol	0.00% (**0**/247)	0.00% (**0**/324)	0.00% (**0**/96)	0.00% (**0**/6)
contracts/lib/math/PRBMathSD59×18.sol	0.00% (**0**/183)	0.00% (**0**/362)	0.00% (**0**/57)	0.00% (**0**/22)
contracts/mock/MockMultisigWallet.sol	100.00% (**16**/16)	100.00% (**20**/20)	71.43% (**5**/7)	100.00% (**4**/4)
contracts/mock/MockToken.sol	100.00% (**6**/6)	100.00% (**6**/6)	50.00% (**1**/2)	100.00% (**4**/4)
contracts/mock/MockUSDT.sol	0.00% (**0**/3)	0.00% (**0**/3)	100.00% (**0**/0)	0.00% (**0**/3)
contracts/ustb/UStb.sol	92.16% (**47**/51)	93.91% (**108**/115)	53.33% (**16**/30)	85.71% (**12**/14)
contracts/ustb/UStbMinting.sol	90.86% (**179**/197)	87.18% (**238**/273)	63.49% (**40**/63)	81.25% (**39**/48)
script/DeploymentUtils.sol	0.00% (**0**/37)	0.00% (**0**/51)	0.00% (**0**/7)	0.00% (**0**/10)
script/ustb/DeployUStb.sol	0.00% (**0**/25)	0.00% (**0**/27)	0.00% (**0**/7)	0.00% (**0**/2)
script/ustb/DeployUStbMinting.sol	0.00% (**0**/23)	0.00% (**0**/30)	100.00% (**0**/0)	0.00% (**0**/2)
test/foundry/UStbBaseSetup.sol	100.00% (**43**/43)	100.00% (**44**/44)	100.00% (**1**/1)	100.00% (**1**/1)
test/foundry/UStbMinting.utils.sol	100.00% (**29**/29)	100.00% (**30**/30)	100.00% (**0**/0)	0.00% (**0**/4)
test/foundry/UStbMintingBaseSetup.sol	99.47% (**186**/187)	99.01% (**201**/203)	100.00% (**3**/3)	42.86% (**3**/7)
test/utils/SigUtils.sol	33.33% (**1**/3)	20.00% (**1**/5)	100.00% (**0**/0)	33.33% (**1**/3)
test/utils/Utils.sol	0.00% (**0**/11)	0.00% (**0**/16)	100.00% (**0**/0)	0.00% (**0**/3)
Total	29.36% (**539**/1836)	26.49% (**688**/2597)	10.46% (**70**/669)	38.21% (**81**/212)

Changelog

- 2024-10-28 - Initial report
- 2024-10-31 - Final report

About Quantstamp

Quantstamp is a global leader in blockchain security. Founded in 2017, Quantstamp's mission is to securely onboard the next billion users to Web3 through its best-in-class Web3 security products and services.

Quantstamp's team consists of cybersecurity experts hailing from globally recognized organizations including Microsoft, AWS, BMW, Meta, and the Ethereum Foundation. Quantstamp engineers hold PhDs or advanced computer science degrees, with decades of combined experience in formal verification, static analysis, blockchain audits, penetration testing, and original leading-edge research.

To date, Quantstamp has performed more than 500 audits and secured over $200 billion in digital asset risk from hackers. Quantstamp has worked with a diverse range of customers, including startups, category leaders and financial institutions. Brands that Quantstamp has worked with include Ethereum 2.0, Binance, Visa, PayPal, Polygon, Avalanche, Curve, Solana, Compound, Lido, MakerDAO, Arbitrum, OpenSea and the World Economic Forum.

Quantstamp's collaborations and partnerships showcase our commitment to world-class research, development and security. We're honored to work with some of the top names in the industry and proud to secure the future of web3.

Notable Collaborations & Customers:

- Blockchains: Ethereum 2.0, Near, Flow, Avalanche, Solana, Cardano, Binance Smart Chain, Hedera Hashgraph, Tezos
- DeFi: Curve, Compound, Maker, Lido, Polygon, Arbitrum, SushiSwap
- NFT: OpenSea, Parallel, Dapper Labs, Decentraland, Sandbox, Axie Infinity, Illuvium, NBA Top Shot, Zora
- Academic institutions: National University of Singapore, MIT

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication or other making available of the report to you by Quantstamp.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites&aspo; owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on any website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any output generated by such software.

Disclaimer

The review and this report are provided on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Quantstamp disclaims all warranties, expressed implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. You agree that access and/or use of the report and other results of the review, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE. This report is based on the scope of materials and documentation provided for a limited review at the time provided. You acknowledge that Blockchain technology remains under development and is subject to unknown risks and flaws and, as such, the report may not be complete or inclusive of all vulnerabilities. The review is limited to the materials identified in the report and does not extend to the compiler layer, or any other areas beyond the programming language, or programming aspects that could present security risks. The report does not indicate the endorsement by Quantstamp of any particular project or team, nor guarantee its security, and and may not be represented as such. No third party is entitled to rely on the report in any any way, including for the purpose of making any decisions to buy or sell a product, product, service or any other asset. Quantstamp does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or or any open source or third-party software, code, libraries, materials, or information to, to, called by, referenced by or accessible through the report, its content, or any related related services and products, any hyperlinked websites, or any other websites or mobile applications, and we will not be a party to or in any way be responsible for monitoring any any transaction between you and any third party. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.

Quantstamp