# Shell Protocol Audit

| | |
|---|---|
| **Date** | June 2020 |
| **Lead Auditor** | Daniel Luca |
| **Co-auditors** | Gonçalo Sá |

# 1 Executive Summary

This report presents the results of our engagement with **Counterparty** to review **Shell Protocol**.

The review was conducted over the course of two weeks and two days, **from June 22 to July 7 2020** by Daniel Luca and Gonçalo Sá. A total of **22** person-days were spent. This review is following another previous 1-day review we provided for the client.

During the **first week**, we started to learn how the system works by having a few calls with the client and by reading the provided documents and the available source code. We set up a meeting with the development team on Monday to explain our process, understand the system, agree on the overall scope and purpose of the audit, and ask for a commit hash.

We had some initial problems with compiling the code because we were not familiar with the framework, and not all of the changes were pushed to the repository. Over the next few days, we went back and forth, trying to identify the problems and coming up with solutions on how to make the code compilable. Once we agreed on a pretty good version, we locked in the commit hash `2f0f9d6c5a6ba471ae88f14da1f1b3e8470332b0`. A complete list of files can be found in the Appendix.

On Tuesday, we had another call with the development team to discuss the high-level overview of the system, roughly getting into the math behind the balancing mechanics. We also asked for a walk-through of the system, to understand how a user is supposed to interact with it. We got familiar with the core functionality of the system, namely, how the balancing is done and how tokens flow within the system. We also discussed why some decisions were made as they are, specifically how internal accounting is done, how external calls are done, and why they exist, how the assimilators work in principle.

On Wednesday, we continued to do our manual review and set up a new meeting with the client to discuss initial findings, ask questions, and continue the code walk-through. Our main focus was to follow the lifetime of a simple token swap transaction throughout the codebase.

Over the following days, we directed our efforts towards understanding the system, trying to find issues and edge cases. A number of issues were filed to be included in the report.

It was clear to us that it is vital to help the development team change the way they are currently developing the application. A number of systemic problems were identified and explained in detail below, in the Action Items section.

On Friday, we set up another meeting with the client where we discussed that our focus would be split into finding more security issues with the system, but also address the overall methodology of the development process.

In the **second week** we continued to file issues while categorizing them into groups relating to *complexity*, *fragility*, and *testing*. Each of the groups is referred to in the Action Items. Other security issues not relating to the aforementioned groups were filed in the review.

We continued to have daily sync meetings to discuss any issues we might have found and use our group knowledge to validate and test different attack vectors. Most of the

time spent in the second week was to file new issues, validate attack vectors, and put the report together in a format that the audit team believes to deliver the most value to the development team.

At the end of the second week, we had a meeting with the development team where we discussed a few other attack vectors, power centralization issues, and other parts of the code that were not completely clear.

### 1.1 Mitigations Review Update

The Shell Protocol team diligently addressed all of the issues present in the report. This effort entailed huge code transformations that were completed at a fast pace.

The biggest transformation was the creation of a “pool partitioning” mechanism to tackle the lock-up conditions stemming from the pool balancing loop needed for the correct functioning of the system.

Since the beginning of the audit, there were clear improvements in both the quality of the code style and the attention to the product’s security.

The auditing team would also like to make notice of the fact that the codebase was in a developing state by the time of the audit and therefore strong changes were sure to ensue.

## 2 Scope

Our review focused on the commit hash `2f0f9d6c5a6ba471ae88f14da1f1b3e8470332b0`. The list of files in scope can be found in the Appendix.

### 2.1 Objectives

Together with the **Counterparty** team, we identified the following priorities for our review:

1. Ensure that the system is implemented consistently with the intended functionality and without unintended edge cases.
2. Identify known vulnerabilities particular to smart contract systems, as outlined in our Smart Contract Best Practices, and the Smart Contract Weakness Classification Registry.
3. Try to find ways to reduce gas costs.

# 3 System Overview

Shell Protocol is a conjunction of a liquidity pool and AMM for stablecoins that is designed to have no slippage beyond the liquidity fee and to pass arbitrage profits on to the liquidity providers (LPs, from now on). To achieve this goal, Shell Protocol implements a bonding surface in its core logic made up of several smaller, locally-defined bonding surfaces.

Even though Shell Protocol is made of clearly delineated business logic modules, the codebase under review implements them in a way heavily intertwined way. As such, it is easier to distinguish and categorize these components based on their logical functions rather than specific files or contracts.

`Loihi` is the name of this version of Shell Protocol's codebase and the one we will be describing in the next few paragraphs.

The two main logical components of the system are `shell` s and `assimilators`. With a `shell` being the most central part of the system and the `assimilators` being the middleware connecting the different financial instruments to the core liquidity and swap logic for each pool.

A formalism of the bonding surface implemented by `Loihi` was provided to the audit team and can be found here.

## 3.1 Shell

A shell is, as stated, the core, logical part of Shell Protocol. Each instance of a pool will have exactly one shell.

The shell is the data structure present within the smart contract system that ties all the other components together. Encapsulated inside the resulting compiled smart contract that makes up a shell there is logic for:

- the core mathematical components for the loss function creating the bonding curve ( `Shells.sol` , `Loihi.sol` )
- ERC20 implementation of the shell token ( `ShellsExternal.sol` , `Shells.sol` )
- depositing liquidity in the pool ( `Loihi.sol` )
- withdrawing liquidity from the pool ( `Loihi.sol` )
- swapping two tokens supported by the pool ( `Loihi.sol` )

- administrative functions that rule the pool's parameters ( `Controller.sol` , `Loihi.sol` , `LoihiRoot.sol` )

Most of the files mentioned above take the form of all-internal-method libraries that get fully appended to `Loihi.sol` 's and `LoihiRoot.sol` 's bytecode, since they are the only contracts in the set.

In addition to the components, Shell Protocol is using safe math libraries to handle both 256-bit unsigned integer and 128-bit 64.64 fixed-point decimal arithmetics with no over/underflow.

All the mathematical components handling the fixed fee, halt enforcement parameters and slippage calculations are using 64.64 fixed-point decimals, internally, using Solidity's elementary 128-bit signed integer type.

## Liquidity-related And Swap-related Components

All the math-related methods are contained within `Shells.sol` with some amount of preparation for the calculations being done inside `Loihi.sol` , as well.

In `Loihi.sol` , the system prepares the data that is fed into the implementation of the bonding surface in `Shells.sol` . Loihi queries the token balances for the set of supported stablecoins and their derivatives, creates the necessary derived variables needed for the calculations (mostly sums of all the token balances, through the methods `getLiquidityData` and `getSwapData` ) and then calls the relevant methods for the calculations ( `calculateLiquidityMembrane` and `calculateTrade` , related to the prior methods in the respective order) from `Shells.sol` .

The Loihi contract is then responsible for propagating the changes (and effectively writing them to storage) to the `omega` parameter of the pool and calling the respective assimilator methods to credit or withdraw the relevant amount of tokens while implementing boundary checks for these values at the same time.

## Administrative Functions

Most of the administrative functions logic is implemented in `Controller.sol` with some of it being implemented in `Loihi.sol` (more specifically the ability to remove an assimilator and the ability to send tokens from the contract to an address of their choice).

The functions present in the Controller allow the administrator to set new parameters for the pool, include a new supported asset (stablecoin) and include a new assimilator for any supported asset.

## 3.2 Assimilators

The assimilators are the middleware between a shell and the different DeFi systems it needs to interact in its set of supported assets and their derivatives.

They act, in essence, as a *delegatecall* proxy system to all the different stablecoins and their derivatives in order for the pool to be able to interactively balance itself and allow LPs to provide liquidity.

In the current architecture, assimilators necessarily need to take the shape of proxies to externally deployed contracts. This is due to the fact that each of the supported assets and its derivatives have differently named methods and even control flows that need to be followed in order to interact with their token implementations properly.

Not only are assimilators an abstraction to the different interfaces and accounting models of each of the supported assets and derivatives but a necessary instrument in the normalization of each of these external tokens' value internally to Shell Protocol.

In the file `Assimilators.sol`, all the methods present are merely internal functions meant to delegate execution to the each relevant implementation (given the relevant token) at runtime. The critical part of the assimilator architecture is present in the `assimilators/` directory inside the repository under review.

The actual implementations of the assimilators (only meant to be *delegatecalled*) mostly implement the same interface and are the components responsible for both interfacing with the external DeFi systems and also make the correctness checks about the success of these necessary sub-calls. Assimilators are also keeping the consistency by typecasting between the conventional unsigned integer used for balances in ERC20-compatible tokens and the 64.64 fixed-point decimal used by the shell, internally.

# 4 Action Items

## 4.1 Reduce overall complexity

### Mitigations Review Update

*Comment from the development team*:

Previously, we were utilizing libraries with the “using library for type” convention. This made the code difficult to understand.

Now our library use is well named and with the exception of mathematical operations is employed using the normal call syntax like “Library.function(argument, argument)”. Combined with descriptive names for the libraries, it is easy to see where the code is flowing to.

Although we now make use of a total of 9 non-math libraries (including internal and external libraries), they are well named and easy to reason with.

---

Complexity comes at the cost of security. Complex systems are harder to understand, harder to test, and harder to maintain.

For smart contract systems, the fault-intolerant environment of the EVM necessarily demands that security is the highest priority. Therefore, it should be a design goal of all smart contract systems to reduce complexity and make logic explicit wherever possible.

The Shell Protocol is a highly complex system:

- There is a deep library usage that spans between multiple files, even having libraries include other libraries a few levels deep.
- The mathematical model is not completely and clearly defined; the document explaining the math powering the system has not reached a final version.
- A large number of assimilators can be included as part of the system. Each of them has to be reviewed in the context of the system, but also in the context of the token being supported because adding an assimilator which is incorrectly implemented can put the system at risk.
- Fixed point math operations are used in the system, but the libraries were changed, and some of the implementations are duplicated in multiple contracts.
- A common theme throughout the system is to use `delegatecall`, which creates a huge trust issue since the owner can, at any point in time, add an assimilator that steals all the tokens in the system.
- There are inconsistencies in function names and variable names; these should all be addressed. For example “Assimilators” used to be called “Adapters”, and some

of the function names still refer to "Adapters", we have `includeAssimilator` and `excludeAdapter`.

### Recommendation:

Reducing overall complexity is no simple task, and addressing this system's complexity will require careful thought and consideration outside of the scope of this review. In general, prioritize the following concepts:

- **Optimize for readability.** Ensure that code is as easy to understand as possible. Implement clear and consistent naming conventions, group similar functions within the same file, and generally attempt to structure and organize the code so that humans can read and understand it best.
- **Remove commented-out code.** Remove old code that was used for tests or for setting up local environments and find other ways to mock or configure the system without the need to change code.

### Related:

| Issues | Priority |
|---|---|
| Remove commented out code from the repository | High |
| Remove debugging code from the repository | Medium |
| Use consistent interfaces for functions in the same group | Medium |
| Use one file for each contract or library | Low |

## 4.2 Increase the overall quality and quantity of testing

### Mitigations Review Update

*Comment from the development team*:

The failing tests existed because we made minute changes to our present model (changes in applying the base fee - "epsilon"), so in a sense, rather than failing they just need updating. Many of them are also an artifact of architecting the tests in such a way that they can be run against arbitrary parameter sets - or in different "suites".

---

Several findings of this assessment suggest that Shell Protocol is inadequately tested:

- Almost half of the tests fail.
- There is no continuous integration system.
- There is no report about code coverage. We do not know if the tests cover the whole codebase. This makes it likely that not all functionality is well tested.
- Some of the changes implemented in the fork libraries do not implement the intended functionality.

### Recommendation:

Implementing a robust, complete test suite requires careful consideration outside of the scope of this review. In general, prioritize the following concepts:

- **Write tests that encapsulate the specification.** Tests should address each of a system's requirements. A system's requirements should be clearly defined within the system design specification.
- **Try to develop new functionality by writing tests first.** Test-driven development makes sure that all of the written code is accompanied by a test.
- **Implement a continuous integration system.** Using one of the platforms that offer CI/CD services and implements a list of actions that do compilation, tests, code coverage, and panics when the smallest piece does not check out.

### Related:

| Issues | Priority |
|---|---|
| Tests should not fail | High |
| Code coverage should be close to 100% | Medium |

## 4.3 Address codebase fragility

### Mitigations Review Update

*Comment from the development team*:

---

Software is considered "fragile" when issues or changes in one part of the system can have side-effects in conceptually unrelated parts of the codebase. Fragile software tends to break easily and may be challenging to maintain.

Our assessment uncovered that for each swap in the system, all of the enabled assets run code. That means that if one of the enabled tokens blacklists the exchange, all of the tokens are locked in the system unless a backdoor exists.

### Recommendation:

Building an anti-fragile system requires careful thought and consideration outside of the scope of this review. In general, prioritize the following concepts:

- **Follow checks-effects-interactions pattern.** The checks-effects-interactions should be implemented throughout the code. Also, functions' return values should always be checked for correctness.
- **Follow the single-responsibility principle of functions.** This principle states that functions should have responsibility for a single part of the system's functionality and that their purpose should be narrowly-aligned with that responsibility. Avoid functions that "do everything" (like `setGovernanceParameter`), and avoid functions that touch every other function (like `funding` and `markPrice`).

### Related:

| Issues | Priority |
|---|---|
| Functions do not check the correctness of the arguments | High |
| Math library's fork has problematic changes | Medium |
| Check return values for both internal and external calls | Medium |

# 5 Security Specification

This section describes, **from a security perspective**, the expected behavior of the system under audit. It is not a substitute for documentation. The purpose of this section is to identify specific security properties that were validated by the audit team.

## 5.1 Actors

The relevant actors are listed below with their respective abilities:

- Non-privileged access actors
  - `Pool user` (i.e., non-privileged user with no shell tokens in their possession)
  - Can swap assets supported by the pool.

    - `Liquidity provider`
    - All of the above.
    - Can deposit supported assets into the pool.
    - Can withdraw its share of supported assets from the pool (relative to the amount of shell tokens they hold).
- Privileged access actors
    - `Administrator`
    - All of the above.
    - Setting all the parameters of the pool at anytime.
    - Adding supported assets.
    - Adding supported assimilators (basically setting an address to which execution is delegated, no restrictions).

## 5.2 Important Security Properties

The following is a non-exhaustive list of security properties that were verified in this audit:

- Non-privileged access actors
    - `Pool user`
    - Cannot swap assets that are unsupported by the pool.
    - Cannot swap an asset bypassing the fee calculation.
    - Cannot bypass the depositing of the *intake* token.
    - `Liquidity provider`
    - Cannot bypass the fee calculation when depositing or withdrawing assets.
    - Cannot mint or burn tokens in a proportion not relative to their held shell tokens.
        - By repeated action, cannot drain the pool by exploiting a bad implementation of the pre-fee-calculation parameters.

Please note that some other properties were reviewed in addition to these ones. The ones that were proven to be untrue are instead represented as issues in this same report.

# 6 Issues

Each issue has an assigned severity:

- Minor issues are subjective in nature. They are typically suggestions around best practices or readability. Code maintainers should use their own judgment as to whether to address such issues.
- Medium issues are objective in nature but are not security vulnerabilities. These should be addressed unless there is a clear reason not to.
- Major issues are security vulnerabilities that may not be directly exploitable or may require certain conditions in order to be exploited. All major issues should be addressed.
- Critical issues are directly exploitable security vulnerabilities that need to be fixed.

## 6.1 Unexpected response in an assimilator's external call can lock-up the whole system Major ✓ Fixed

### Resolution

*Comment from the development team*:

When this was brought to our attention, it made the most sense to look at it from a bird's eye view. In the event that an assimilator does seize up either due to smart contract malfunctioning or to some type of governance decision in one of our dependencies, then depending on the severity of the event, it could either make it so that that particular dependency is unable to be transacted with or it could brick the pool altogether.

In the case of the latter severity where the pool is bricked altogether for an extended period of time, then this means the end of that particular pool's life. In this case, we find it prudent to allow for the withdrawal of any asset still functional from the pool. Should such an event transpire, we have instituted functionality to allow users to withdraw individually from the pool's assets according to their Shell balances without being exposed to the inertia of the incapacitated assets.

In such an event, the owner of the pool can now trigger a partitioned state which is an end of life state for the pool in which users send Shells as normal until they decide to redeem any portion of them, after which they will only be able to

redeem the portion of individual asset balances their Shell balance held claims on.

## Description

The assimilators, being the "middleware" between a shell and all the external DeFi systems it interacts with, perform several external calls within their methods, as would be expected.

An example of such a contract is `mainnetSUsdToASUsdAssimilator.sol` (the contract can be found here).

The problem outlined in the title arises from the fact that Solidity automatically checks for the successful execution of the underlying message call (i.e., it bubbles up assertions and reverts) and, therefore, if any of these external systems changes in unexpected ways the call to the shell will revert itself.

This problem is immensely magnified by the fact that *all* the external methods in `Loihi` dealing with deposits, withdraws, and swaps rebalance the pool and, as a consequence, all of the assimilators for the reserve tokens get called at some point.

In summary, if any of the reserve tokens start, for some reason, refusing to complete a call to some of their methods, the whole protocol stops working, and the tokens are locked in forever (this is assuming the development team removes the `safeApprove` function from `Loihi`, v. issue 6.3).

## Recommendation

There is no easy solution to this problem since calls to these external systems cannot simply be ignored. Shell needs successful responses from the reserve assimilators to be able to function properly.

One possible mitigation is to create a trustless mechanism based on repeated misbehavior by an external system to be able to remove a reserve asset from the pool.

Such a design could consist of an external function accessible to all actors that needs *X* confirmations over a period of *Y* blocks (or days, for that matter) with even spacing between them to be able to remove a reserve asset.

This means that no trust to the owners is implied (since this would require the extreme power to take user's tokens) and still maintains the healthy option of being able to remove faulty tokens from the pool.

## 6.2 Certain functions lack input validation routines Major ✓ Fixed

### Resolution

*Comment from the development team*:

1. Now all functions in the Orchestrator revert on incorrect arguments.
2. All functions in Loihi in general revert on incorrect arguments.

### Description

The functions should first check if the passed arguments are valid first. The checks-effects-interactions pattern should be implemented throughout the code.

These checks should include, but not be limited to:

- `uint` should be larger than `0` when `0` is considered invalid
- `uint` should be within constraints
- `int` should be positive in some cases
- length of arrays should match if more arrays are sent as arguments
- addresses should not be `0x0`

### Examples

The function `includeAsset` does not do any checks before changing the contract state.

**src/Loihi.sol:L59-L61**

```
function includeAsset (address _numeraire, address _nAssim, address _reserve, address
    shell.includeAsset(_numeraire, _nAssim, _reserve, _rAssim, _weight);
}
```

The internal function called by the public method `includeAsset` again doesn't check any of the data.

**src/Controller.sol:L77-L97**

```
function includeAsset (Shells.Shell storage shell, address _numeraire, address _numer

    Assimilators.Assimilator storage _numeraireAssimilator = shell.assimilators[_nume

    _numeraireAssimilator.addr = _numeraireAssim;

    _numeraireAssimilator.ix = uint8(shell.numeraires.length);

    shell.numeraires.push(_numeraireAssimilator);

    Assimilators.Assimilator storage _reserveAssimilator = shell.assimilators[_reserv

    _reserveAssimilator.addr = _reserveAssim;

    _reserveAssimilator.ix = uint8(shell.reserves.length);

    shell.reserves.push(_reserveAssimilator);

    shell.weights.push(_weight.divu(1e18).add(uint256(1).divu(1e18)));

}
```

Similar with `includeAssimilator`.

**src/Loihi.sol:L63-L65**

```
function includeAssimilator (address _numeraire, address _derivative, address _assimi
    shell.includeAssimilator(_numeraire, _derivative, _assimilator);
}
```

Again no checks are done in any function.

**src/Controller.sol:L99-L106**

```
function includeAssimilator (Shells.Shell storage shell, address _numeraire, address

    Assimilators.Assimilator storage _numeraireAssim = shell.assimilators[_numeraire]

    shell.assimilators[_derivative] = Assimilators.Assimilator(_assimilator, _numerai
    // shell.assimilators[_derivative] = Assimilators.Assimilator(_assimilator, _numera

}
```

Not only does the administrator functions not have any checks, but also user facing functions do not check the arguments.

For example `swapByOrigin` does not check any of the arguments if you consider it calls `MainnetDaiToDaiAssimilator`.

**src/Loihi.sol:L85-L89**

```
function swapByOrigin (address _o, address _t, uint256 _oAmt, uint256 _mTAmt, uint256

    return transferByOrigin(_o, _t, _dline, _mTAmt, _oAmt, msg.sender);

}
```

It calls `transferByOrigin` and we simplify this example and consider we have `_o.ix == _t.ix`

**src/Loihi.sol:L181-L187**

```
function transferByOrigin (address _origin, address _target, uint256 _dline, uint256

    Assimilators.Assimilator memory _o = shell.assimilators[_origin];
    Assimilators.Assimilator memory _t = shell.assimilators[_target];

    // TODO: how to include min target amount
    if (_o.ix == _t.ix) return _t.addr.outputNumeraire(_rcpnt, _o.addr.intakeRaw(_oAm
```

In which case it can call 2 functions on an assimilatior such as `MainnetDaiToDaiAssimilator`.

The first called function is `intakeRaw`.

**src/assimilators/mainnet/daiReserves/mainnetDaiToDaiAssimilator.sol:L42-L49**

```
// transfers raw amonut of dai in, wraps it in cDai, returns numeraire amount
function intakeRaw (uint256 _amount) public returns (int128 amount_, int128 balance_)

    dai.transferFrom(msg.sender, address(this), _amount);

    amount_ = _amount.divu(1e18);

}
```

And its result is used in `outputNumeraire` that again does not have any checks.

**src/assimilators/mainnet/daiReserves/mainnetDaiToDaiAssimilator.sol:L83-L92**

```
// takes numeraire amount of dai, unwraps corresponding amount of cDai, transfers that
function outputNumeraire (address _dst, int128 _amount) public returns (uint256 amour

    amount_ = _amount.mulu(1e18);

    dai.transfer(_dst, amount_);

    return amount_;

}
```

### Recommendation

Implement the `checks-effects-interactions` as a pattern to write code. Add tests that check if all of the arguments have been validated.

Consider checking arguments as an important part of writing code and developing the system.

## 6.3 Remove `Loihi` methods that can be used as backdoors by the administrator Major ✓ Fixed

| Resolution |
| --- |
| Issue was partly addressed by the development team. However, the feature to add new assimilators is still present and that ultimately means that the administrators have power to run arbitrary bytecode. |

*Updated remediation response* Since the development team still hadn't fully settled on a strategy for a mainnet launch, this was left as a residue even after the audit mitigation phase. However, at launch time, this issue was no longer present and all the assimilators are now defined at deploy-time, it is fully resolved.

## Description

There are several functions in `Loihi` that give extreme powers to the shell administrator. The most dangerous set of those is the ones granting the capability to add assimilators.

Since assimilators are essentially a proxy architecture to delegate code to several different implementations of the same interface, the administrator could, intentionally or unintentionally, deploy malicious or faulty code in the implementation of an assimilator. This means that the administrator is essentially totally trusted to not run code that, for example, drains the whole pool or locks up the users' and LPs' tokens.

In addition to these, the function `safeApprove` allows the administrator to move any of the tokens the contract holds to any address regardless of the balances any of the users have.

This can also be used by the owner as a backdoor to completely drain the contract.

**src/Loihi.sol:L643-L649**

```
function safeApprove(address _token, address _spender, uint256 _value) public onlyOwr

    (bool success, bytes memory returndata) = _token.call(abi.encodeWithSignature("ap

    require(success, "SafeERC20: low-level call failed");

}
```

## Recommendation

Remove the `safeApprove` function and, instead, use a trustless escape-hatch mechanism like the one suggested in issue 6.1.

For the assimilator addition functions, our recommendation is that they are made completely internal, only callable in the constructor, at deploy time.

Even though this is not a big structural change (in fact, it *reduces* the attack surface), it is, indeed, a feature loss. However, this is the only way to make each shell a time-invariant system.

This would not only increase Shell's security but also would greatly improve the trust the users have in the protocol since, after deployment, the code is now **static** and auditable.

## 6.4 Assimilators should implement an interface Major ✓ Fixed

| Resolution |
| --- |
| *Comment from the development team*:<br><br>They now implement the interface in "src/interfaces/IAssimilator.sol". |

### Description

The Assimilators are one of the core components within the application. They are used to move the tokens and can be thought of as a "middleware" between the Shell Protocol application and any other supported tokens.

The methods attached to the assimilators are called throughout the application and they are a critical component of the whole system. Because of this fact, it is extremely important that they behave correctly.

A suggestion to restrict the possibility of errors when implementing them and when using them is to make all of the assimilators implement a unique specific interface. This way, any deviation would be immediately observed, right when the compilation happens.

### Examples

Consider this example. The user calls `swapByOrigin`.

**src/Loihi.sol:L85-L89**

```
function swapByOrigin (address _o, address _t, uint256 _oAmt, uint256 _mTAmt, uint256

    return transferByOrigin(_o, _t, _dline, _mTAmt, _oAmt, msg.sender);

}
```

Which calls `transferByOrigin`. In `transferByOrigin`, if the origin index matches the target index, a different execution branch is activated.

**src/Loihi.sol:L187**

```
if (_o.ix == _t.ix) return _t.addr.outputNumeraire(_rcpnt, _o.addr.intakeRaw(_oAmt));
```

In this case we need the output of `_o.addr.intakeRaw(_oAmt)`.

If we pick a random assimilator and check the implementation, we see the function `intakeRaw` needs to return the transferred amount.

**src/assimilators/mainnet/daiReserves/mainnetCDaiToDaiAssimilator.sol:L52-L67**

```
// takes raw cdai amount, transfers it in, calculates corresponding numeraire amount an
function intakeRaw (uint256 _amount) public returns (int128 amount_) {

    bool success = cdai.transferFrom(msg.sender, address(this), _amount);

    if (!success) revert("CDai/transferFrom-failed");

    uint256 _rate = cdai.exchangeRateStored();

    _amount = ( _amount * _rate ) / 1e18;

    cdai.redeemUnderlying(_amount);

    amount_ = _amount.divu(1e18);

}
```

However, with other implementations, the returns do not match. In the case of `MainnetDaiToDaiAssimilator`, it returns 2 values, which will make the `Loihi` contract work in this case but can misbehave in other cases, or even fail.

**src/assimilators/mainnet/daiReserves/mainnetDaiToDaiAssimilator.sol:L42-L49**

```
// transfers raw amonut of dai in, wraps it in cDai, returns numeraire amount
function intakeRaw (uint256 _amount) public returns (int128 amount_, int128 balance_)

    dai.transferFrom(msg.sender, address(this), _amount);

    amount_ = _amount.divu(1e18);

}
```

Making all the assimilators implement one unique interface will enforce the functions to look the same from the outside.

### Recommendation

Create a unique interface for the assimilators and make all the contracts implement that interface.

## 6.5 Assimilators do not conform to the ERC20 specification Medium

✓ Fixed

### Resolution

*Comment from the development team*:

All calls to compliant ERC20s now check for return booleans. Non compliant ERC20s are called with a function that checks for the success of the call.

### Description

The assimilators in the codebase make heavy usage of both the `transfer` and `transferFrom` methods in the ERC20 standard.

Quoting the relevant parts of the specification of the standard:

> Transfers _value amount of tokens to address _to, and MUST fire the Transfer event. The function SHOULD throw if the message caller's account balance does not have enough tokens to spend.

> The transferFrom method is used for a withdraw workflow, allowing contracts to transfer tokens on your behalf. This can be used for example to allow a contract to transfer tokens on your behalf and/or to charge fees in sub-currencies. The function SHOULD throw unless the _from account has deliberately authorized the sender of the message via some mechanism.

We can see that, even though it is suggested that ERC20-compliant tokens do `throw` on the lack of authorization from the sender or lack of funds to complete the transfer, the standard does not enforce it.

This means that, in order to make the system both more resilient and future-proof, code in each implementation of current and future assimilators should check for the return value of both `transfer` and `transferFrom` call instead of just relying on the external contract to revert execution.

The extent of this issue is only mitigated by the fact that new assets are only added by the shell administrator and could, therefore, be audited prior to their addition.

### Non-exhaustive Examples

**src/assimilators/mainnet/daiReserves/mainnetDaiToDaiAssimilator.sol:L45**

```
dai.transferFrom(msg.sender, address(this), _amount);
```

**src/assimilators/mainnet/daiReserves/mainnetDaiToDaiAssimilator.sol:L64**

```
dai.transfer(_dst, _amount);
```

### Recommendation

Add a check for the return boolean of the function.

Example:

```
require(dai.transferFrom(msg.sender, address(this), _amount) == true);
```

## 6.6 Access to assimilators does not check for existence and allows delegation to the zeroth address Medium ✓ Fixed

## Resolution

*Comment from the development team*:

All retrieval of assimilators now check that the assimilators address is not the zeroth address.

## Description

For every method that allows to selectively withdraw, deposit, or swap tokens in `Loihi`, the user is allowed to specify addresses for the assimilators of said tokens (by inputting the addresses of the tokens themselves).

The shell then performs a lookup on a mapping called `assimilators` inside its main structure and uses the result of that lookup to delegate call the assimilator deployed by the shell administrator.

However, there are no checks for prior instantiation of a specific, supported token, effectively meaning that we can do a lookup on an all-zeroed-out member of that mapping and delegate call execution to the zeroth address.

The only thing preventing execution from going forward in this unwanted fashion is the fact that the ABI decoder expects a certain return data size from the interface implemented in `Assimilator.sol`.

For example, the 32 bytes expected as a result of this call:

**src/Assimilators.sol:L58-L66**

```
function viewNumeraireAmount (address _assim, uint256 _amt) internal returns (int128

    // amount_ = IAssimilator(_assim).viewNumeraireAmount(_amt); // for production

    bytes memory data = abi.encodeWithSelector(iAsmltr.viewNumeraireAmount.selector,

    amt_ = abi.decode(_assim.delegate(data), (int128)); // for development

}
```

This is definitely an insufficient check since the interface for the assimilators might change in the future to include functions that have no return values.

### Recommendation

Check for the prior instantiation of assimilators by including the following requirement:

```
require(shell.assimilators[<TOKEN_ADDRESS>].ix != 0);
```

In all the functions that access the `assimilators` mapping and change the indexes to be 1-based instead pf 0-based.

## 6.7 Math library's fork has problematic changes Medium ✓ Fixed

### Description

The math library ABDK Libraries for Solidity was forked and modified to add a few `unsafe_*` functions.

- `unsafe_add`
- `unsafe_sub`
- `unsafe_mul`
- `unsafe_div`
- `unsafe_abs`

The problem which was introduced is that `unsafe_add` ironically is not really unsafe, it is as safe as the original `add` function. It is, in fact, identical to the safe `add` function.

**src/ABDKMath64x64.sol:L102-L113**

```
/**
 * Calculate x + y.  Revert on overflow.
 *
 * @param x signed 64.64-bit fixed point number
 * @param y signed 64.64-bit fixed point number
 * @return signed 64.64-bit fixed point number
 */
function add (int128 x, int128 y) internal pure returns (int128) {
  int256 result = int256(x) + y;
  require (result >= MIN_64x64 && result <= MAX_64x64);
  return int128 (result);
}
```

**src/ABDKMath64x64.sol:L115-L126**

```
/**
 * Calculate x + y.  Revert on overflow.
 *
 * @param x signed 64.64-bit fixed point number
 * @param y signed 64.64-bit fixed point number
 * @return signed 64.64-bit fixed point number
 */
function unsafe_add (int128 x, int128 y) internal pure returns (int128) {
  int256 result = int256(x) + y;
  require (result >= MIN_64x64 && result <= MAX_64x64);
  return int128 (result);
}
```

Fortunately, `unsafe_add` is not used anywhere in the code.

However, `unsafe_abs` was changed from this:

**src/ABDKMath64x64.sol:L322-L331**

```
/**
 * Calculate |x|.  Revert on overflow.
 *
 * @param x signed 64.64-bit fixed point number
 * @return signed 64.64-bit fixed point number
 */
function abs (int128 x) internal pure returns (int128) {
  require (x != MIN_64x64);
  return x < 0 ? -x : x;
}
```

To this:

**src/ABDKMath64x64.sol:L333-L341**

```
/**
 * Calculate |x|.  Revert on overflow.
 *
 * @param x signed 64.64-bit fixed point number
 * @return signed 64.64-bit fixed point number
 */
function unsafe_abs (int128 x) internal pure returns (int128) {
  return x < 0 ? -x : x;
}
```

The check that was removed, is actually an important check:

```
require (x != MIN_64x64);
```

**src/ABDKMath64x64.sol:L19**

```
int128 private constant MIN_64x64 = -0x80000000000000000000000000000000;
```

The problem is that for an `int128` variable that is equal to `-0x80000000000000000000000000000000`, there is no absolute value within the constraints of `int128`.

Starting from `int128 n = -0x80000000000000000000000000000000`, the absolute value should be `int128 abs_n = -n`, however `abs_n` is equal to the initial value of `n`. The final value of `abs_n` is still `-0x80000000000000000000000000000000`. It's still not a positive or zero value. The operation `0 - n` wraps back to the same initial value.

### Recommendation

Remove unused `unsafe_*` functions and try to find other ways of doing unsafe math (if it is fundamentally important) without changing existing, trusted, already audited code.

## 6.8 Use one file for each contract or library Medium ✓ Fixed

| Resolution |
| --- |
| Issue fixed by the development team. |

## Description

The repository contains a lot of contracts and libraries that are added in the same file as another contract or library.

Organizing the code in this manner makes it hard to navigate, develop and audit. It is a best practice to have each contract or library in its own file. The file also needs to bear the name of the hosted contract or library.

## Examples

### src/Shells.sol:L20

```
library SafeERC20Arithmetic {
```

### src/Shells.sol:L32

```
library Shells {
```

### src/Loihi.sol:L26-L28

```
contract ERC20Approve {
    function approve (address spender, uint256 amount) public returns (bool);
}
```

### src/Loihi.sol:L30

```
contract Loihi is LoihiRoot {
```

### src/Assimilators.sol:L19

```
library Delegate {
```

**src/Assimilators.sol:L33**

```
library Assimilators {
```

### Recommendation

Split up contracts and libraries in single files.

## 6.9 Remove debugging code from the repository Medium ✓ Fixed

| Resolution |
| --- |
| Issue fixed but he development team. |

### Description

Throughout the repository, there is source code from the development stage that was used for debugging the functionality and was not removed.

This should not be present in the source code and even if they are used while functionality is developed, they should be removed after the functionality was implemented.

### Examples

**src/Shells.sol:L63-L67**

```
event log(bytes32);
event log_int(bytes32, int256);
event log_ints(bytes32, int256[]);
event log_uint(bytes32, uint256);
event log_uints(bytes32, uint256[]);
```

**src/Assimilators.sol:L44-L46**

```
event log(bytes32);
event log_uint(bytes32, uint256);
event log_int(bytes32, int256);
```

### src/Controller.sol:L33-L37

```
event log(bytes32);
event log_int(bytes32, int128);
event log_int(bytes32, int);
event log_uint(bytes32, uint);
event log_addr(bytes32, address);
```

### src/LoihiRoot.sol:L53

```
event log(bytes32);
```

### src/Shells.sol:L63-L67

```
event log(bytes32);
event log_int(bytes32, int256);
event log_ints(bytes32, int256[]);
event log_uint(bytes32, uint256);
event log_uints(bytes32, uint256[]);
```

### src/Loihi.sol:L470-L474

```
event log_int(bytes32, int);
event log_ints(bytes32, int128[]);
event log_uint(bytes32, uint);
event log_uints(bytes32, uint[]);
event log_addrs(bytes32, address[]);
```

### src/assimilators/mainnet/cdaiReserves/mainnetDaiToCDaiAssimilator.sol:L35-L36

```
event log_uint(bytes32, uint256);
event log_int(bytes32, int256);
```

### src/assimilators/mainnet/cusdcReserves/mainnetUsdcToCUsdcAssimilator.sol:L38

```
event log_uint(bytes32, uint256);
```

### src/Loihi.sol:L51

```
shell.testHalts = true;
```

**src/LoihiRoot.sol:L79-L83**

```
function setTestHalts (bool _testOrNotToTest) public {

    shell.testHalts = _testOrNotToTest;

}
```

**src/Shells.sol:L60**

```
bool testHalts;
```

### Recommendation

Remove the debug functionality at the end of the development cycle of each functionality.

## 6.10 Tests should not fail Medium ✓ Fixed

### Resolution

*Comment from the development team*:

The failing tests are because we made minute changes to our present model (changes in applying the base fee - "epsilon"), so in a sense, rather than failing they just need updating. Many of them are also an artifact of architecting the tests in such a way that they can be run against arbitrary parameter sets - or in different "suites".

### Description

The role of the tests should be to make sure the application behaves properly. This should include positive tests (functionality that should be implemented) and negative

tests (behavior stopped or limited by the application).

The test suite should pass 100% of the tests. After spending time with the development team, we managed to ask for the changes that allowed us to run the tests suite. This revealed that out of the 555 tests, 206 are failing. This staggering number does not allow us to check what the problem is and makes anybody running tests ignore them completely.

Tests should be an integral part of the codebase, and they should be considered as important (or even more important) than the code itself. One should be able to recreate the whole codebase by just having the tests.

### Recommendation

Update tests in order for the whole of the test suite to pass.

## 6.11 Remove commented out code from the repository Medium

✓ Fixed

### Description

Having commented out code increases the cognitive load on an already complex system. Also, it hides the important parts of the system that should get the proper attention, but that attention gets to be diluted.

There is no code that is important enough to be left commented out in a repository. Git branching should take care of having different code versions or diffs should show what was before.

If there is commented out code, this also has to be maintained; it will be out of date if other parts of the system are changed, and the tests will not pick that up.

The main problem is that commented code adds confusion with no real benefit. Code should be code, and comments should be comments.

### Examples

Commented out code should be removed or dealt with in a separate branch that is later included in the master branch.

**src/Assimilators.sol:L48-L56**

```
function viewRawAmount (address _assim, int128 _amt) internal returns (uint256 amount

    // amount_ = IAssimilator(_assim).viewRawAmount(_amt); // for production

    bytes memory data = abi.encodeWithSelector(iAsmltr.viewRawAmount.selector, _amt.a

    amount_ = abi.decode(_assim.delegate(data), (uint256)); // for development

}
```

### src/Assimilators.sol:L58-L66

```
function viewNumeraireAmount (address _assim, uint256 _amt) internal returns (int128

    // amount_ = IAssimilator(_assim).viewNumeraireAmount(_amt); // for production

    bytes memory data = abi.encodeWithSelector(iAsmltr.viewNumeraireAmount.selector,

    amt_ = abi.decode(_assim.delegate(data), (int128)); // for development

}
```

### src/Assimilators.sol:L58-L66

```
function viewNumeraireAmount (address _assim, uint256 _amt) internal returns (int128

    // amount_ = IAssimilator(_assim).viewNumeraireAmount(_amt); // for production

    bytes memory data = abi.encodeWithSelector(iAsmltr.viewNumeraireAmount.selector,

    amt_ = abi.decode(_assim.delegate(data), (int128)); // for development

}
```

### src/Controller.sol:L99-L106

```
function includeAssimilator (Shells.Shell storage shell, address _numeraire, address

    Assimilators.Assimilator storage _numeraireAssim = shell.assimilators[_numeraire]

    shell.assimilators[_derivative] = Assimilators.Assimilator(_assimilator, _numerai
    // shell.assimilators[_derivative] = Assimilators.Assimilator(_assimilator, _numera

}
```

### src/Loihi.sol:L596-L618

```
function transfer (address _recipient, uint256 _amount) public nonReentrant returns (
    // return shell.transfer(_recipient, _amount);
}

function transferFrom (address _sender, address _recipient, uint256 _amount) public r
    // return shell.transferFrom(_sender, _recipient, _amount);
}

function approve (address _spender, uint256 _amount) public nonReentrant returns (boo
    // return shell.approve(_spender, _amount);
}

function increaseAllowance(address _spender, uint256 _addedValue) public returns (boo
    // return shell.increaseAllowance(_spender, _addedValue);
}

function decreaseAllowance(address _spender, uint256 _subtractedValue) public returns
    // return shell.decreaseAllowance(_spender, _subtractedValue);
}

function balanceOf (address _account) public view returns (uint256) {
    // return shell.balances[_account];
}
```

### src/test/deposits/suiteOne.t.sol:L15-L29

```
// function test_s1_selectiveDeposit_noSlippage_balanced_10DAI_10USDC_10USDT_2p5SUSD_NO

//     uint256 newShells = super.noSlippage_balanced_10DAI_10USDC_10USDT_2p5SUSD();

//     assertEq(newShells, 32499999216641686631);

// }

// function test_s1_selectiveDeposit_noSlippage_balanced_10DAI_10USDC_10USDT_2p5SUSD_HA

//     uint256 newShells = super.noSlippage_balanced_10DAI_10USDC_10USDT_2p5SUSD_HACK()

//     assertEq(newShells, 32499999216641686631);

// }
```

### src/test/deposits/depositsTemplate.sol:L40-L56

```
// function noSlippage_balanced_10DAI_10USDC_10USDT_2p5SUSD_HACK () public returns (uin

//     uint256 startingShells = l.proportionalDeposit(300e18);

//     uint256 gas = gasleft();

//     shellsMinted_ = l.depositHack(
//         address(dai), 10e18,
//         address(usdc), 10e6,
//         address(usdt), 10e6,
//         address(susd), 2.5e18
//     );

//     emit log_uint("gas for deposit", gas - gasleft());


// }
```

### Recommendation

Remove all the commented out code or transform it into comments.

## 6.12 Should check if the asset already exists when adding a new asset Medium ✓ Fixed

### Resolution

*Comment from the development team*:

We have decided not to have dynamic adding/removing of assets in this release.

### Description

The public function `includeAsset`

**src/Loihi.sol:L128-L130**

```
function includeAsset (address _numeraire, address _nAssim, address _reserve, address
    shell.includeAsset(_numeraire, _nAssim, _reserve, _rAssim, _weight);
}
```

Calls the internal `includeAsset` implementation

**src/Controller.sol:L72**

```
function includeAsset (Shells.Shell storage shell, address _numeraire, address _numer
```

But there is no check to see if the asset already exists in the list. Because the check was not done, `shell.numeraires` can contain multiple identical instances.

**src/Controller.sol:L80**

```
shell.numeraires.push(_numeraireAssimilator);
```

### Recommendation

Check if the `_numeraire` already exists before invoking `includeAsset`.

## 6.13 Check return values for both internal and external calls Minor

✓ Fixed

### Resolution

*Comment from the development team*:

This doesn't seem feasible. Checking how much was transferred to/from the contract would pose unacceptable gas costs. Because of these constraints, the value returned by the assimilator methods never touches the outside world. They are just converted into numeraire format and returned, so checking these values would not provide any previously unknown information.

### Description

There are some cases where functions which return values are called throughout the source code but the return values are not processed, nor checked.

The returns should in principle be handled and checked for validity to provide more robustness to the code.

## Examples

The function `intakeNumeraire` receives a number of tokens and returns how many tokens were transferred to the contract.

**src/assimilators/mainnet/daiReserves/mainnetDaiToDaiAssimilator.sol:L51-L59**

```
// transfers numeraire amount of dai in, wraps it in cDai, returns raw amount
function intakeNumeraire (int128 _amount) public returns (uint256 amount_) {

    // truncate stray decimals caused by conversion
    amount_ = _amount.mulu(1e18) / 1e3 * 1e3;

    dai.transferFrom(msg.sender, address(this), amount_);

}
```

Similarly, the function `outputNumeraire` receives a destination address and an amount of token for withdrawal and returns a number of transferred tokens to the specified address.

**src/assimilators/mainnet/daiReserves/mainnetDaiToDaiAssimilator.sol:L83-L92**

```
// takes numeraire amount of dai, unwraps corresponding amount of cDai, transfers that
function outputNumeraire (address _dst, int128 _amount) public returns (uint256 amour

    amount_ = _amount.mulu(1e18);

    dai.transfer(_dst, amount_);

    return amount_;

}
```

However, the results are not handled in the main contract.

**src/Loihi.sol:L497**

```
shell.numeraires[i].addr.intakeNumeraire(_shells.mul(shell.weights[i]));
```

**src/Loihi.sol:L509**

```
shell.numeraires[i].addr.intakeNumeraire(_oBals[i].mul(_multiplier));
```

**src/Loihi.sol:L586**

```
shell.reserves[i].addr.outputNumeraire(msg.sender, _oBals[i].mul(_multiplier));
```

A sanity check can be done to make sure that more than 0 tokens were transferred to the contract.

```
unit intakeAmount = shell.numeraires[i].addr.intakeNumeraire(_shells.mul(shell.weight
require(intakeAmount > 0, "Must intake a positive number of tokens");
```

### Recommendation

Handle all return values everywhere returns exist and add checks to make sure an expected value was returned.

If the return values are never used, consider not returning them at all.

## 6.14 Interfaces do not need to be implemented for the compiler to access their selectors. Minor ✓ Fixed

### Resolution

*Comment from the development team*:

This is the case for the version we used, solc 0.5.15. Versions 0.5.17 and 0.6.* do not require it.

### Description

In `Assimilators.sol` the interface for the assimilators is implemented in a state variable constant as an interface to the zeroth address in order to make use of it's selectors.

**src/Assimilators.sol:L37**

```
IAssimilator constant iAsmltr = IAssimilator(address(0));
```

This pattern is unneeded since you can reference selectors by using the imported interface directly without any implementation. It hinders both gas costs and readability of the code.

### Examples

### Recommendation

Delete line 37 in `Assimilators.sol` and instead of getting selectors through:

**src/Assimilators.sol:L62**

```
bytes memory data = abi.encodeWithSelector(iAsmltr.viewNumeraireAmount.selector, _amt
```

use the expression:

`IAssimilator.viewRawAmount.selector`

## 6.15 Use consistent interfaces for functions in the same group

Minor ✓ Fixed

### Description

In the file `Shells.sol`, there also is a library that is being used internally for safe adds and subtractions.

This library has 2 functions.

`add` which receives 2 arguments, `x` and `y`.

**src/Shells.sol:L22-L24**

```
function add(uint x, uint y) internal pure returns (uint z) {
    require((z = x + y) >= x, "add-overflow");
}
```

`sub` which receives 3 arguments `x`, `y` and `_errorMessage`.

**src/Shells.sol:L26-L28**

```
function sub(uint x, uint y, string memory _errorMessage) internal pure returns (uint
    require((z = x - y) <= x, _errorMessage);
}
```

In order to reduce the cognitive load on the auditors and developers alike, somehow-related functions should have coherent logic and interfaces. Both of the functions either need to have 2 arguments, with an implied error message passed to `require`, or both functions need to have 3 arguments, with an error message that can be specified.

### Recommendation

Update the functions to be coherent with other related functions.

## 6.16 Code coverage should be close to 100% Minor ✓ Fixed

### Resolution

*Comment from the development team*:

This is true for all aspects of the bonding curve.

Things that have been tested on Kovan with the frontend dapp but could use a unit test are things relevant to sending shell tokens - issuing approvals, transfers and transferfroms.

The adding of assets and assimilators are tested by proxy because they are dependencies for the entire behavior of the bonding surface.

For this release, we plan on having the assets and the assimilators frozen at launch, so there is not much to test regarding continuous updating/changing of assets and assimilators.

We have, however, considered allowing for the dynamic adjustment of weights in addition to parameters.

### Description

Code coverage is a measure used to describe how much of the source code is executed during the automated test suite. A system with high code coverage, measured as lines of code executed, has a lower chance to contain undiscovered bugs.

The codebase does not have any information about the code coverage.

### Recommendation

Make the test suite output code coverage and add more tests to handle the lines of code that are not touched by any tests.

## 6.17 Consider emitting an event when changing the frozen state of the contract Minor ✓ Fixed

### Description

The function `freeze` allows the owner to freeze and unfreeze the contract.

**src/Loihi.sol:L144-L146**

```
function freeze (bool _freeze) public onlyOwner {
    frozen = _freeze;
}
```

The common pattern when doing actions important for the outside of the blockchain is to emit an event when the action is successful.

It's probably a good idea to emit an event stating the contract was frozen or unfrozen.

### Recommendation

Create an event that displays the current state of the contract.

```
event Frozen(bool frozen);
```

And emit the event when `frozen` is called.

```
function freeze (bool _freeze) public onlyOwner {
    frozen = _freeze;
    emit Frozen(_freeze);
}
```

## 6.18 Function `supportsInterface` can be restricted to `pure`

Minor ✓ Fixed

### Description

The function `supportsInterface` returns a `bool` stating that the contract supports one of the defined interfaces.

**src/Loihi.sol:L140-L142**

```
function supportsInterface (bytes4 interfaceID) public returns (bool) {
    return interfaceID == ERC20ID || interfaceID == ERC165ID;
}
```

The function does not access or change the state of the contract, this is why it can be restricted to `pure`.

### Recommendation

Restrict the function definition to `pure`.

```
function supportsInterface (bytes4 interfaceID) public pure returns (bool) {
```

## 6.19 Use more consistent function naming (includeAssimilator / excludeAdapter) Minor ✓ Fixed

### Description

The function `includeAssimilator` adds a new assimilator to the list

**src/Controller.sol:L98**

```
shell.assimilators[_derivative] = Assimilators.Assimilator(_assimilator, _numeraireAs
```

The function `excludeAdapter` removes the specified assimilator from the list

**src/Loihi.sol:L137**

```
delete shell.assimilators[_assimilator];
```

### Recommendation

Consider renaming the function `excludeAdapter` to `removeAssimilator` and moving the logic of adding and removing in the same source file.

# Appendix 1 - Files in Scope

This audit covered the following files:

| File Name | SHA-1 Hash |
|---|---|
| src/Assimilators.sol | 3f6cc11fc01be7d858de29255ff2dcd7c73535a3 |
| src/Controller.sol | 96fefe583cf31c7ef45f2094367ae1527ed1fa3e |
| src/Loihi.sol | de9feda8b31fae8494b5ea995d898be3251431a2 |
| src/LoihiRoot.sol | e2b21cdab22c7a42cc7ff03e5b202d67cc6c8d04 |
| src/Shells.sol | 2ae89c49fcec7d83aef5f7f0d95bd9e17d9efacb |
| src/ShellsExternal.sol | becc7634a4bf45d08060be2fcb5e01382b6f8d4f |
| src/assimilators/AssimilatorMath.sol | c4dfe2367edb23dab938d50d57a17fd5bb4c94b2 |
| src/assimilators/aaveResources/ILendingPool.sol | fe26c09c3be97a5bb37de95aa4ae895c948da251 |

| File Name | SHA-1 Hash |
| --- | --- |
| src/assimilators/aaveResources/ILendingPoolAddressesProvider.sol | 0f845e0d8d8456a963ce2717bdbccf27f58a4bf2 |
| src/assimilators/mainnet/asusdReserves/mainnetASusdToASusdAssimilator.sol | e1d56000137d13db62abafbc240a24f943ace70b |
| src/assimilators/mainnet/asusdReserves/mainnetSUsdToASUsdAssimilator.sol | 35e8dbbb137e70a36598a8f783e396d9e8d0e5c5 |
| src/assimilators/mainnet/ausdtReserves/mainnetAUsdtToAUsdtAssimilator.sol | ea16a1544b169760821fed668bebee52ef99e72b |
| src/assimilators/mainnet/ausdtReserves/mainnetUsdtToAUsdtAssimilator.sol | 495334fcb505cc45a628ca332438feb66183c772 |
| src/assimilators/mainnet/cdaiReserves/mainnetCDaiToCDaiAssimilator.sol | c268af639fe6e862917a5995ab1045222b325a03 |
| src/assimilators/mainnet/cdaiReserves/mainnetChaiToCDaiAssimilator.sol | ab2dc7613ac8b0dd4a44b19b643fc4c650711694 |
| src/assimilators/mainnet/cdaiReserves/mainnetDaiToCDaiAssimilator.sol | 0794a05a05356c73575da70ffa30d595ca53162f |
| src/assimilators/mainnet/cusdcReserves/mainnetCUsdcToCUsdcAssimilator.sol | 4fbc3fc0b9fe2117460741bfeff80e80252afa51 |
| src/assimilators/mainnet/cusdcReserves/mainnetUsdcToCUsdcAssimilator.sol | 88e582d815fc08d34de014827a2ff4ec93f29292 |
| src/assimilators/mainnet/daiReserves/mainnetCDaiToDaiAssimilator.sol | 4a7d6eec1e609eb94590e2c37db80fc4fc5ea4ab |
| src/assimilators/mainnet/daiReserves/mainnetChaiToDaiAssimilator.sol | 17b65aa02c02bf7ae98b2ca6d78892a99890ddb9 |
| src/assimilators/mainnet/daiReserves/mainnetDaiToDaiAssimilator.sol | 531f1c5c2982267eedcb9b52fa9f5dc611f5ae49 |
| src/assimilators/mainnet/susdReserves/MainnetASusdToSUsdAssimilator.sol | 444ff56afc3c610179976dcf56cbb8e6ce3029c0 |

| File Name | SHA-1 Hash |
| --- | --- |
| src/assimilators/mainnet/susdReserves/MainnetSUsdToSUsdAssimilator.sol | d33559f600a9a0a46c76a23b97c23b62a117c687 |
| src/assimilators/mainnet/usdcReserves/localCUsdcToUsdcAssimilator.sol | a63719169882e86a86620e3a505ef1e62f05d71c |
| src/assimilators/mainnet/usdcReserves/localUsdcToUsdcAssimilator.sol | 95c5bc3b9470c74b0cc34a97c7f504d6ecb68033 |
| src/assimilators/mainnet/usdtReserves/localAUsdtToUsdtAssimilator.sol | 27c8b33955a9d6f043ab9ff9ff66fa9a916b0bc1 |
| src/assimilators/mainnet/usdtReserves/localUsdtToUsdtAssimilator.sol | 3da930b6f8d30210405e9f71189f2250b52a6287 |

# Appendix 2 - Artifacts

This section contains some of the artifacts generated during our review by automated tools, the test suite, etc. If any issues or recommendations were identified by the output presented here, they have been addressed in the appropriate section above.

## A.2.1 MythX

MythX is a security analysis API for Ethereum smart contracts. It performs multiple types of analysis, including fuzzing and symbolic execution, to detect many common vulnerability types. The tool was used for automated vulnerability discovery for all audited contracts and libraries. More details on MythX can be found at mythx.io.

The PDF report of the initial MythX vulnerability scan can be found here.

The PDF report for the followup MythX vulnerability scan, after code changes, can be found here.

## A.2.2 Ethlint

Ethlint is an open source project for linting Solidity code. Only security-related issues were reviewed by the audit team.

Below is the raw output of the Ethlint vulnerability scan:

Click to expand Ethlint output

```
src/Assimilators.sol
  17:0     warning    "pragma solidity >0.4.13;" should be at the top of the file.
  23:60    warning    Avoid using low-level function 'delegatecall'.
  25:8     error      Avoid using Inline Assembly.

src/Controller.sol
  39:4    warning    Line exceeds the limit of 145 characters    max-len
  77:4    warning    Line exceeds the limit of 145 characters    max-len

src/Loihi.sol
  181:4      warning    Line exceeds the limit of 145 characters            max-len
  271:4      warning    Line exceeds the limit of 145 characters            max-len
  345:32     error      Only use indent of 16 spaces.                       indentat
  433:4      warning    Line exceeds the limit of 145 characters            max-len
  434:16     warning    Avoid using 'block.timestamp'.                      security
  526:4      warning    Line exceeds the limit of 145 characters            max-len
  527:16     warning    Avoid using 'block.timestamp'.                      security
  573:23     error      Only use indent of 8 spaces.                        indentat
  596:95     warning    Code contains empty block                           no-empty
  600:116    warning    Code contains empty block                           no-empty
  604:101    warning    Code contains empty block                           no-empty
  608:101    warning    Code contains empty block                           no-empty
  612:106    warning    Code contains empty block                           no-empty
  616:72     warning    Code contains empty block                           no-empty
  624:88     warning    Code contains empty block                           no-empty
  645:23     error      Variable 'returndata' is declared but never used.   no-unuse
  645:57     warning    Avoid using low-level function 'call'.              security

src/LoihiRoot.sol
  40:1     warning    Line contains trailing whitespace           no-trailing-whitesp
  67:16    error      Only use indent of 4 spaces.                indentation
  67:30    error      Only use indent of 4 spaces.                indentation
  67:42    error      Only use indent of 4 spaces.                indentation
  68:17    error      Only use indent of 4 spaces.                indentation
  69:23    error      Only use indent of 4 spaces.                indentation
  70:17    error      Only use indent of 4 spaces.                indentation
  72:4     warning    Line exceeds the limit of 145 characters    max-len
  73:20    error      Only use indent of 8 spaces.                indentation
  73:34    error      Only use indent of 8 spaces.                indentation
  73:48    error      Only use indent of 8 spaces.                indentation
  74:22    error      Only use indent of 8 spaces.                indentation
  75:22    error      Only use indent of 8 spaces.                indentation
  76:22    error      Only use indent of 8 spaces.                indentation
```

```
src/Shells.sol
  18:0     warning    "pragma solidity >0.4.13;" should be at the top of the file.
  129:4    error      "calculateTrade": Avoid assigning to function parameters.
  129:4    error      "calculateTrade": Avoid assigning to function parameters.

src/assimilators/aaveResources/ILendingPool.sol
  12:4    warning    Line exceeds the limit of 145 characters    max-len
  14:4    warning    Line contains trailing whitespace           no-trailing-whitespa

src/assimilators/kovan/kovanASUsdAssimilator.sol
  26:26    warning    Code contains empty block                                    n
  41:4     error      "intakeNumeraire": Avoid assigning to function parameters.    s
  55:4     error      "outputNumeraire": Avoid assigning to function parameters.    s
  88:4     warning    Line contains trailing whitespace                            n
  94:8     warning    Provide an error message for require()                       e

src/assimilators/kovan/kovanAUsdtAssimilator.sol
  13:26    warning    Code contains empty block                                    n
  28:4     error      "intakeNumeraire": Avoid assigning to function parameters.    s
  41:4     error      "outputNumeraire": Avoid assigning to function parameters.    s
  74:4     warning    Line contains trailing whitespace                            n
  80:8     warning    Provide an error message for require()                       e

src/assimilators/kovan/kovanCDaiAssimilator.sol
  20:26    warning    Code contains empty block                 no-empty-blocks
  85:4     warning    Line contains trailing whitespace         no-trailing-whitespac
  91:8     warning    Provide an error message for require()    error-reason

src/assimilators/kovan/kovanCUsdcAssimilator.sol
  21:26    warning    Code contains empty block
  24:4     warning    Line contains trailing whitespace
  29:4     warning    Line contains trailing whitespace
  31:4     error      "intakeNumeraire": Avoid assigning to function parameters.
  40:4     error      "outputNumeraire": Avoid assigning to function parameters.
  54:4     error      "viewRawAmount": Avoid assigning to function parameters.
  76:26    warning    Single space should be either on both sides of '/' or not at al
  76:41    warning    There should be no whitespace or comments between argument and

src/assimilators/kovan/kovanChaiAssimilator.sol
  30:26    warning    Code contains empty block                           no-empty-blo
  66:4     warning    Line contains trailing whitespace                   no-trailing-
  86:20    warning    Avoid using 'now' (alias to 'block.timestamp').    security/no-
  91:20    warning    Avoid using 'now' (alias to 'block.timestamp').    security/no-
  93:0     error      Only use indent of 8 spaces.                        indentation
  101:4    warning    Line contains trailing whitespace                   no-trailing-
  107:8    warning    Provide an error message for require()              error-reason

src/assimilators/kovan/kovanDaiAssimilator.sol
  22:26    warning    Code contains empty block            no-empty-blocks
  30:8     warning    Line contains trailing whitespace    no-trailing-whitespace
  33:8     warning    Line contains trailing whitespace    no-trailing-whitespace
  39:8     warning    Line contains trailing whitespace    no-trailing-whitespace
```

```
  43:8     warning    Line contains trailing whitespace    no-trailing-whitespace
  49:8     warning    Line contains trailing whitespace    no-trailing-whitespace
  52:8     warning    Line contains trailing whitespace    no-trailing-whitespace
  56:8     warning    Line contains trailing whitespace    no-trailing-whitespace
  60:8     warning    Line contains trailing whitespace    no-trailing-whitespace
  88:8     warning    Line contains trailing whitespace    no-trailing-whitespace
  90:8     warning    Line contains trailing whitespace    no-trailing-whitespace

src/assimilators/kovan/kovanSUsdAssimilator.sol
  23:26     warning    Code contains empty block
  45:4      error      "intakeNumeraire": Avoid assigning to function parameters.
  65:4      error      "outputNumeraire": Avoid assigning to function parameters.
  99:4      warning    Line contains trailing whitespace
  110:23    error      Variable 'returndata' is declared but never used.
  110:56    warning    Avoid using low-level function 'call'.

src/assimilators/kovan/kovanUsdcAssimilator.sol
  21:26    warning    Code contains empty block                                    r
  35:4     error      "intakeNumeraire": Avoid assigning to function parameters.    s
  48:4     error      "outputNumeraire": Avoid assigning to function parameters.    s

src/assimilators/kovan/kovanUsdtAssimilator.sol
  23:26     warning    Code contains empty block
  45:4      error      "intakeNumeraire": Avoid assigning to function parameters.
  63:4      error      "outputNumeraire": Avoid assigning to function parameters.
  95:4      warning    Line contains trailing whitespace
  106:23    error      Variable 'returndata' is declared but never used.
  106:56    warning    Avoid using low-level function 'call'.

src/assimilators/local/ausdtReserves/localUsdtToAUsdtAssimilator.sol
  33:0      error      Only use indent of 4 spaces.              indentation
  125:56    warning    Avoid using low-level function 'call'.    security/no-low-leve
  126:8     error      Avoid using Inline Assembly.              security/no-inline-a

src/assimilators/local/cdaiReserves/localChaiToCDaiAssimilator.sol
  69:4     error      "intakeRaw": Avoid assigning to function parameters.
  117:4    error      "outputRaw": Avoid assigning to function parameters.
  145:4    error      "viewNumeraireAmount": Avoid assigning to function parameters.
  165:4    warning    Line contains trailing whitespace

src/assimilators/local/cusdcReserves/localCUsdcToCUsdcAssimilator.sol
  77:8    error    Variable '_balanceBefore' is declared but never used.    no-unusec

src/assimilators/local/cusdcReserves/localUsdcToCUsdcAssimilator.sol
  115:8    warning    Line contains trailing whitespace    no-trailing-whitespace

src/assimilators/local/daiReserves/localCDaiToDaiAssimilator.sol
  36:4    error    "intakeRawAndGetBalance": Avoid assigning to function parameters.
  55:4    error    "intakeRaw": Avoid assigning to function parameters.

src/assimilators/local/daiReserves/localChaiToDaiAssimilator.sol
  69:4     error      "intakeRawAndGetBalance": Avoid assigning to function parameter
```

```
  87:4     error      "intakeRaw": Avoid assigning to function parameters.
  123:4    error      "outputRawAndGetBalance": Avoid assigning to function parameter
  138:4    error      "outputRaw": Avoid assigning to function parameters.
  158:4    error      "viewNumeraireAmount": Avoid assigning to function parameters.
  176:4    error      "viewNumeraireAmountAndBalance": Avoid assigning to function pa
  179:8    warning    Line contains trailing whitespace

src/assimilators/local/usdcReserves/localCUsdcToUsdcAssimilator.sol
  37:4    error    "intakeRawAndGetBalance": Avoid assigning to function parameters.
  56:4    error    "intakeRaw": Avoid assigning to function parameters.

src/assimilators/local/usdtReserves/localUsdtToUsdtAssimilator.sol
  30:4      warning    Line contains trailing whitespace         no-trailing-whitespa
  136:56    warning    Avoid using low-level function 'call'.    security/no-low-leve
  137:8     error      Avoid using Inline Assembly.              security/no-inline-a

src/assimilators/mainnet/asusdReserves/mainnetASusdToASusdAssimilator.sol
  30:26    warning    Code contains empty block    no-empty-blocks

src/assimilators/mainnet/asusdReserves/mainnetSUsdToASUsdAssimilator.sol
  38:26    warning    Code contains empty block    no-empty-blocks

src/assimilators/mainnet/ausdtReserves/mainnetAUsdtToAUsdtAssimilator.sol
  32:26    warning    Code contains empty block    no-empty-blocks

src/assimilators/mainnet/ausdtReserves/mainnetUsdtToAUsdtAssimilator.sol
  34:26     warning    Code contains empty block                 no-empty-blocks
  158:56    warning    Avoid using low-level function 'call'.    security/no-low-leve
  159:8     error      Avoid using Inline Assembly.              security/no-inline-a

src/assimilators/mainnet/cdaiReserves/mainnetCDaiToCDaiAssimilator.sol
  30:26    warning    Code contains empty block    no-empty-blocks

src/assimilators/mainnet/cdaiReserves/mainnetChaiToCDaiAssimilator.sol
  39:26    warning    Code contains empty block
  42:4     error      "intakeRaw": Avoid assigning to function parameters.
  59:4     error      "intakeRawAndGetBalance": Avoid assigning to function parameter
  110:4    error      "outputRaw": Avoid assigning to function parameters.
  125:4    error      "outputRawAndGetBalance": Avoid assigning to function parameter

src/assimilators/mainnet/cdaiReserves/mainnetDaiToCDaiAssimilator.sol
  33:26    warning    Code contains empty block    no-empty-blocks

src/assimilators/mainnet/cusdcReserves/mainnetCUsdcToCUsdcAssimilator.sol
  30:26    warning    Code contains empty block    no-empty-blocks

src/assimilators/mainnet/cusdcReserves/mainnetUsdcToCUsdcAssimilator.sol
  36:26    warning    Code contains empty block    no-empty-blocks

src/assimilators/mainnet/daiReserves/mainnetCDaiToDaiAssimilator.sol
  25:4     warning    Line contains trailing whitespace
  29:26    warning    Code contains empty block
```

```
  32:4     error      "intakeRawAndGetBalance": Avoid assigning to function parameter
  53:4     error      "intakeRaw": Avoid assigning to function parameters.

src/assimilators/mainnet/daiReserves/mainnetChaiToDaiAssimilator.sol
  31:26    warning    Code contains empty block
  65:4     error      "intakeRawAndGetBalance": Avoid assigning to function parameter
  83:4     error      "intakeRaw": Avoid assigning to function parameters.
  119:4    error      "outputRawAndGetBalance": Avoid assigning to function parameter
  134:4    error      "outputRaw": Avoid assigning to function parameters.
  154:4    error      "viewNumeraireAmount": Avoid assigning to function parameters.
  172:4    error      "viewNumeraireAmountAndBalance": Avoid assigning to function pa

src/assimilators/mainnet/daiReserves/mainnetDaiToDaiAssimilator.sol
  27:26    warning    Code contains empty block    no-empty-blocks

src/assimilators/mainnet/susdReserves/MainnetASusdToSUsdAssimilator.sol
  32:26    warning    Code contains empty block    no-empty-blocks

src/assimilators/mainnet/susdReserves/MainnetSUsdToSUsdAssimilator.sol
  27:26    warning    Code contains empty block    no-empty-blocks

src/assimilators/mainnet/usdcReserves/localCUsdcToUsdcAssimilator.sol
  29:26    warning    Code contains empty block
  32:4     error      "intakeRawAndGetBalance": Avoid assigning to function parameter
  55:4     error      "intakeRaw": Avoid assigning to function parameters.

src/assimilators/mainnet/usdcReserves/localUsdcToUsdcAssimilator.sol
  27:26    warning    Code contains empty block    no-empty-blocks

src/assimilators/mainnet/usdtReserves/localAUsdtToUsdtAssimilator.sol
  33:26     warning    Code contains empty block                 no-empty-blocks
  164:4     warning    Line contains trailing whitespace         no-trailing-whitespa
  174:56    warning    Avoid using low-level function 'call'.    security/no-low-leve
  175:8     error      Avoid using Inline Assembly.              security/no-inline-a

src/assimilators/mainnet/usdtReserves/localUsdtToUsdtAssimilator.sol
  27:26     warning    Code contains empty block                 no-empty-blocks
  134:56    warning    Avoid using low-level function 'call'.    security/no-low-leve
  135:8     error      Avoid using Inline Assembly.              security/no-inline-a

src/test/continuities/suiteSix.t.sol
  13:4     error      Using 'l' for a variable name should be avoided.    variable-de
  144:1    warning    Line contains trailing whitespace                   no-trailing

src/test/debug.t.sol
  20:4    error      Using 'l' for a variable name should be avoided.    variable-dec
  35:8    error      Variable 'p3divu' is declared but never used.       no-unused-va
  44:4    warning    Line contains trailing whitespace                   no-trailing-
  49:8    error      Variable 'a64' is declared but never used.          no-unused-va

src/test/deposits/depositsTemplate.sol
  18:4     error      Using 'l' for a variable name should be avoided.         variab
```

```
  157:4    warning    Line exceeds the limit of 145 characters                max-le
  431:4    warning    Line exceeds the limit of 145 characters                max-le
  452:4    warning    Line exceeds the limit of 145 characters                max-le
  454:8    error      Variable 'startingShells' is declared but never used.    no-unu
  853:4    warning    Line contains trailing whitespace                       no-tra

src/test/deposits/suiteOne.t.sol
  223:4    warning    Line exceeds the limit of 145 characters    max-len
  231:4    warning    Line exceeds the limit of 145 characters    max-len

src/test/deposits/suiteTwo.t.sol
  207:4    warning    Line exceeds the limit of 145 characters    max-len
  215:4    warning    Line exceeds the limit of 145 characters    max-len

src/test/deposits/views/depositsViewsTemplate.sol
  18:4     error      Using 'l' for a variable name should be avoided.         variab
  157:4    warning    Line exceeds the limit of 145 characters                max-le
  431:4    warning    Line exceeds the limit of 145 characters                max-le
  452:4    warning    Line exceeds the limit of 145 characters                max-le
  454:8    error      Variable 'startingShells' is declared but never used.    no-unu
  853:4    warning    Line contains trailing whitespace                       no-tra

src/test/deposits/views/suiteOneViews.t.sol
  87:4     warning    Line exceeds the limit of 145 characters    max-len
  223:4    warning    Line exceeds the limit of 145 characters    max-len
  231:4    warning    Line exceeds the limit of 145 characters    max-len

src/test/originSwaps/originSwapTemplate.sol
  19:4      error      Using 'l' for a variable name should be avoided.    variable-c
  158:4     warning    Line contains trailing whitespace                   no-trailir
  465:35    warning    Avoid using low-level function 'call'.              security/r
  485:35    warning    Avoid using low-level function 'call'.              security/r
  500:35    warning    Avoid using low-level function 'call'.              security/r
  515:35    warning    Avoid using low-level function 'call'.              security/r

src/test/originSwaps/suiteFive.t.sol
  31:4    warning    Line contains trailing whitespace    no-trailing-whitespace

src/test/originSwaps/suiteTwo.t.sol
  209:85    warning    Visibility modifier "public" should come before other modifier

src/test/originSwaps/views/originSwapViewsTemplate.sol
  19:4      error      Using 'l' for a variable name should be avoided.    variable-c
  158:4     warning    Line contains trailing whitespace                   no-trailir
  465:35    warning    Avoid using low-level function 'call'.              security/r
  485:35    warning    Avoid using low-level function 'call'.              security/r
  500:35    warning    Avoid using low-level function 'call'.              security/r
  515:35    warning    Avoid using low-level function 'call'.              security/r

src/test/setup/assimilators.sol
  52:4    warning    Line contains trailing whitespace    no-trailing-whitespace
```

```
src/test/setup/loihi.sol
  32:7      error    Only use indent of 8 spaces.    indentation
  33:7      error    Only use indent of 8 spaces.    indentation
  34:7      error    Only use indent of 8 spaces.    indentation
  35:7      error    Only use indent of 8 spaces.    indentation
  36:7      error    Only use indent of 8 spaces.    indentation
  38:7      error    Only use indent of 8 spaces.    indentation
  152:35    error    Only use indent of 8 spaces.    indentation
  153:35    error    Only use indent of 8 spaces.    indentation
  154:36    error    Only use indent of 8 spaces.    indentation
  155:36    error    Only use indent of 8 spaces.    indentation
  156:36    error    Only use indent of 8 spaces.    indentation
  169:35    error    Only use indent of 8 spaces.    indentation
  170:35    error    Only use indent of 8 spaces.    indentation
  171:36    error    Only use indent of 8 spaces.    indentation
  172:36    error    Only use indent of 8 spaces.    indentation
  173:36    error    Only use indent of 8 spaces.    indentation

src/test/setup/methods.sol
  78:59     warning    Avoid using low-level function 'delegatecall'.    security/no-
  80:8      error      Avoid using Inline Assembly.                       security/no-
  302:39    warning    Avoid using low-level function 'call'.             security/no-
  331:39    warning    Avoid using low-level function 'call'.             security/no-
  356:39    warning    Avoid using low-level function 'call'.             security/no-
  377:39    warning    Avoid using low-level function 'call'.             security/no-
  596:39    warning    Avoid using low-level function 'call'.             security/no-
  625:39    warning    Avoid using low-level function 'call'.             security/no-
  650:39    warning    Avoid using low-level function 'call'.             security/no-
  671:39    warning    Avoid using low-level function 'call'.             security/no-
  699:39    warning    Avoid using low-level function 'call'.             security/no-
  728:39    warning    Avoid using low-level function 'call'.             security/no-

src/test/setup/mocks/atoken.sol
  41:56    warning    Avoid using low-level function 'call'.    security/no-low-level
  42:8     error      Avoid using Inline Assembly.              security/no-inline-as

src/test/setup/mocks/cdai.sol
  17:4    warning    Line contains trailing whitespace                 no-trailing-wh
  30:8    error      Variable 'balance' is declared but never used.    no-unused-vars

src/test/setup/mocks/chai.sol
  47:8    warning    Provide an error message for require()    error-reason
  51:8    warning    Provide an error message for require()    error-reason
  55:8    warning    Provide an error message for require()    error-reason

src/test/setup/mocks/erc20.sol
  10:6    error    Only use indent of 8 spaces.    indentation

src/test/setup/mocks/erc20NoBool.sol
  8:6     error    Only use indent of 4 spaces.    indentation
  10:0    error    Only use indent of 4 spaces.    indentation
```

```
src/test/setup/mocks/pot.sol
  10:26    warning    Code contains empty block                          no-empty-blo
  15:4     warning    Line contains trailing whitespace                  no-trailing-
  21:15    warning    Avoid using 'now' (alias to 'block.timestamp').    security/no-

src/test/setup/setup.sol
  57:8     warning    Line contains trailing whitespace    no-trailing-whitespace
  121:8    warning    Line contains trailing whitespace    no-trailing-whitespace
  167:8    warning    Line contains trailing whitespace    no-trailing-whitespace

src/test/targetSwaps/suiteFive.t.sol
  31:4    warning    Line contains trailing whitespace    no-trailing-whitespace

src/test/targetSwaps/targetSwapTemplate.sol
  19:4      error      Using 'l' for a variable name should be avoided.    variable-c
  458:35    warning    Avoid using low-level function 'call'.              security/n
  478:35    warning    Avoid using low-level function 'call'.              security/n
  493:35    warning    Avoid using low-level function 'call'.              security/n
  508:35    warning    Avoid using low-level function 'call'.              security/n
  661:4     warning    Line contains trailing whitespace                   no-trailin

src/test/targetSwaps/views/targetSwapViewsTemplate.sol
  19:4      error      Using 'l' for a variable name should be avoided.    variable-c
  458:35    warning    Avoid using low-level function 'call'.              security/n
  478:35    warning    Avoid using low-level function 'call'.              security/n
  493:35    warning    Avoid using low-level function 'call'.              security/n
  508:35    warning    Avoid using low-level function 'call'.              security/n
  661:4     warning    Line contains trailing whitespace                   no-trailin

src/test/testAssimilators.t.sol
  31:26     warning    Code contains empty block
  34:24     error      Variable 'returndata' is declared but never used.
  34:59     warning    Avoid using low-level function 'call'.
  326:64    warning    Code contains empty block
  350:67    warning    'undefined': The first argument must not be preceded by any wh
  350:92    warning    'undefined': The last argument must not be succeeded by any wh
  384:53    warning    Code contains empty block
  388:54    warning    Code contains empty block
  392:53    warning    Code contains empty block
  396:54    warning    Code contains empty block
  423:50    warning    Code contains empty block
  427:51    warning    Code contains empty block
  431:51    warning    Code contains empty block
  435:52    warning    Code contains empty block
  439:53    warning    Code contains empty block
  443:52    warning    Code contains empty block
  447:53    warning    Code contains empty block
  451:52    warning    Code contains empty block
  455:53    warning    Code contains empty block

src/test/withdraws/suiteOne.t.sol
  183:4    warning    Line exceeds the limit of 145 characters    max-len
```

```
src/test/withdraws/suiteTwo.t.sol
  169:4    warning    Line exceeds the limit of 145 characters    max-len
  261:4    warning    Line contains trailing whitespace           no-trailing-whitesp

src/test/withdraws/views/suiteOneViews.t.sol
  183:4    warning    Line exceeds the limit of 145 characters    max-len

src/test/withdraws/views/withdrawViewsTemplate.sol
  18:4     error      Using 'l' for a variable name should be avoided.          varia
  300:8    error      Variable '_startingShells' is declared but never used.    no-ur
  383:4    warning    Line exceeds the limit of 145 characters                  max-l
  478:8    error      Variable 'startingShells' is declared but never used.     no-ur

src/test/withdraws/withdrawTemplate.sol
  18:4     error      Using 'l' for a variable name should be avoided.          varia
  300:8    error      Variable '_startingShells' is declared but never used.    no-ur
  383:4    warning    Line exceeds the limit of 145 characters                  max-l
  478:8    error      Variable 'startingShells' is declared but never used.     no-ur

✖ 109 errors, 177 warnings found.
```

### A.2.3 Surya

Surya is a utility tool for smart contract systems. It provides a number of visual outputs and information about the structure of smart contracts. It also supports querying the function call graph in multiple ways to aid in the manual inspection and control flow analysis of contracts.

Below is a complete list of functions with their visibility and modifiers:

Click to expand Contracts & File Description Table

›

#### Sūrya's Description Report

##### Contracts Description Table

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | **Function Name** | **Visibility** | **Mutability** | **Modifiers** |
| | | | | |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| **Delegate** | Library | | | |
| └ | delegate | Internal 🔒 | 🛑 | |
| | | | | |
| **Assimilators** | Library | | | |
| └ | viewRawAmount | Internal 🔒 | 🛑 | |
| └ | viewNumeraireAmount | Internal 🔒 | 🛑 | |
| └ | viewNumeraireAmountAndBalance | Internal 🔒 | 🛑 | |
| └ | viewNumeraireBalance | Internal 🔒 | 🛑 | |
| └ | intakeRaw | Internal 🔒 | 🛑 | |
| └ | intakeRawAndGetBalance | Internal 🔒 | 🛑 | |
| └ | intakeNumeraire | Internal 🔒 | 🛑 | |
| └ | outputRaw | Internal 🔒 | 🛑 | |
| └ | outputRawAndGetBalance | Internal 🔒 | 🛑 | |
| └ | outputNumeraire | Internal 🔒 | 🛑 | |
| | | | | |
| **Controller** | Library | | | |
| └ | setParams | Internal 🔒 | 🛑 | |
| └ | includeAsset | Internal 🔒 | 🛑 | |
| └ | includeAssimilator | Internal 🔒 | 🛑 | |
| | | | | |
| **ERC20Approve** | Implementation | | | |
| └ | approve | Public ❗ | 🛑 | NO ❗ |
| | | | | |
| **Loihi** | Implementation | LoihiRoot | | |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | | Public ❗ | 🛑 | NO ❗ |
| L | setParams | Public ❗ | 🛑 | onlyOwner |
| L | includeAsset | Public ❗ | 🛑 | onlyOwner |
| L | includeAssimilator | Public ❗ | 🛑 | onlyOwner |
| L | excludeAdapter | External ❗ | 🛑 | onlyOwner |
| L | supportsInterface | Public ❗ | 🛑 | NO ❗ |
| L | freeze | Public ❗ | 🛑 | onlyOwner |
| L | transferOwnership | Public ❗ | 🛑 | onlyOwner |
| L | swapByOrigin | Public ❗ | 🛑 | notFrozen |
| L | getSwapData | Internal 🔒 | 🛑 | |
| L | viewSwapData | Internal 🔒 | 🛑 | |
| L | transferByOrigin | Public ❗ | 🛑 | notFrozen nonReentrant |
| L | prime | Public ❗ | 🛑 | NO ❗ |
| L | viewOriginTrade | Public ❗ | 🛑 | notFrozen |
| L | swapByTarget | Public ❗ | 🛑 | notFrozen |
| L | transferByTarget | Public ❗ | 🛑 | notFrozen nonReentrant |
| L | viewTargetTrade | Public ❗ | 🛑 | notFrozen |
| L | getLiquidityData | Internal 🔒 | 🛑 | |
| L | viewLiquidityData | Internal 🔒 | 🛑 | |
| L | selectiveDeposit | External ❗ | 🛑 | notFrozen nonReentrant |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | viewSelectiveDeposit | External ❗ | 🛑 | notFrozen |
| L | proportionalDeposit | Public ❗ | 🛑 | notFrozen nonReentrant |
| L | selectiveWithdraw | External ❗ | 🛑 | notFrozen nonReentrant |
| L | viewSelectiveWithdraw | External ❗ | 🛑 | notFrozen |
| L | proportionalWithdraw | Public ❗ | 🛑 | nonReentrant |
| L | transfer | Public ❗ | 🛑 | nonReentrant |
| L | transferFrom | Public ❗ | 🛑 | nonReentrant |
| L | approve | Public ❗ | 🛑 | nonReentrant |
| L | increaseAllowance | Public ❗ | 🛑 | NO ❗ |
| L | decreaseAllowance | Public ❗ | 🛑 | NO ❗ |
| L | balanceOf | Public ❗ | | NO ❗ |
| L | totalSupply | Public ❗ | | NO ❗ |
| L | allowance | Public ❗ | | NO ❗ |
| L | totalReserves | Public ❗ | 🛑 | NO ❗ |
| L | safeApprove | Public ❗ | 🛑 | onlyOwner |
| | | | | |
| **LoihiRoot** | Implementation | | | |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | includeTestAssimilatorState | Public ❗ | 🛑 | NO ❗ |
| L | setTestHalts | Public ❗ | 🛑 | NO ❗ |
| | | | | |
| **SafeERC20Arithmetic** | Library | | | |
| L | add | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| | | | | |
| **Shells** | Library | | | |
| L | calculateFee | Internal 🔒 | | |
| L | calculateMicroFee | Internal 🔒 | | |
| L | calculateTrade | Internal 🔒 | 🛑 | |
| L | calculateLiquidityMembrane | Internal 🔒 | 🛑 | |
| L | enforceHalts | Internal 🔒 | 🛑 | |
| L | burn | Internal 🔒 | 🛑 | |
| L | mint | Internal 🔒 | 🛑 | |
| | | | | |
| **ShellsExternal** | Library | | | |
| L | transfer | External ❗ | 🛑 | NO ❗ |
| L | approve | External ❗ | 🛑 | NO ❗ |
| L | transferFrom | External ❗ | 🛑 | NO ❗ |
| L | increaseAllowance | External ❗ | 🛑 | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | decreaseAllowance | External ❗ | 🛑 | NO ❗ |
| L | _transfer | Private 🔐 | 🛑 | |
| L | _approve | Private 🔐 | 🛑 | |
| | | | | |
| **AssimilatorMath** | Library | | | |
| L | add | Internal 🔒 | | |
| L | sub | Internal 🔒 | | |
| L | mul | Internal 🔒 | | |
| L | wmul | Internal 🔒 | | |
| L | rmul | Internal 🔒 | | |
| L | wdiv | Internal 🔒 | | |
| L | rdiv | Internal 🔒 | | |
| L | rdivup | Internal 🔒 | | |
| | | | | |
| **ILendingPool** | Interface | | | |
| L | deposit | External ❗ | 💵 | NO ❗ |
| L | redeemUnderlying | External ❗ | 🛑 | NO ❗ |
| L | borrow | External ❗ | 🛑 | NO ❗ |
| L | repay | External ❗ | 💵 | NO ❗ |
| L | swapBorrowRateMode | External ❗ | 🛑 | NO ❗ |
| L | rebalanceStableBorrowRate | External ❗ | 🛑 | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | setUserUseReserveAsCollateral | External ❗ | 🛑 | NO ❗ |
| L | liquidationCall | External ❗ | 💵 | NO ❗ |
| L | flashLoan | External ❗ | 🛑 | NO ❗ |
| L | getReserveConfigurationData | External ❗ | | NO ❗ |
| L | getReserveData | External ❗ | | NO ❗ |
| L | getUserAccountData | External ❗ | | NO ❗ |
| L | getUserReserveData | External ❗ | | NO ❗ |
| L | getReserves | External ❗ | | NO ❗ |
| | | | | |
| **ILendingPoolAddressesProvider** | Interface | | | |
| L | getLendingPool | External ❗ | | NO ❗ |
| L | setLendingPoolImpl | External ❗ | 🛑 | NO ❗ |
| L | getLendingPoolCore | External ❗ | | NO ❗ |
| L | setLendingPoolCoreImpl | External ❗ | 🛑 | NO ❗ |
| L | getLendingPoolConfigurator | External ❗ | | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | setLendingPoolConfiguratorImpl | External ❗ | 🛑 | NO ❗ |
| L | getLendingPoolDataProvider | External ❗ | | NO ❗ |
| L | setLendingPoolDataProviderImpl | External ❗ | 🛑 | NO ❗ |
| L | getLendingPoolParametersProvider | External ❗ | | NO ❗ |
| L | setLendingPoolParametersProviderImpl | External ❗ | 🛑 | NO ❗ |
| L | getTokenDistributor | External ❗ | | NO ❗ |
| L | setTokenDistributor | External ❗ | 🛑 | NO ❗ |
| L | getFeeProvider | External ❗ | | NO ❗ |
| L | setFeeProviderImpl | External ❗ | 🛑 | NO ❗ |
| L | getLendingPoolLiquidationManager | External ❗ | | NO ❗ |
| L | setLendingPoolLiquidationManager | External ❗ | 🛑 | NO ❗ |
| L | getLendingPoolManager | External ❗ | | NO ❗ |
| L | setLendingPoolManager | External ❗ | 🛑 | NO ❗ |
| L | getPriceOracle | External ❗ | | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | setPriceOracle | External ❗ | 🛑 | NO ❗ |
| L | getLendingRateOracle | External ❗ | | NO ❗ |
| L | setLendingRateOracle | External ❗ | 🛑 | NO ❗ |
| | | | | |
| **MainnetASUsdToASUsdAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | getASUsd | Public ❗ | | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | | NO ❗ |
| | | | | |
| **MainnetSUsdToASUsdAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | getASUsd | Public ❗ | | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | | NO ❗ |
| | | | | |
| **MainnetAUsdtToAUsdtAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | getAUsdt | Private 🔐 | | |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | viewRawAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | | NO ❗ |
| | | | | |
| **MainnetUsdtToAUsdtAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | getAUsdt | Public ❗ | | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | | NO ❗ |
| L | safeTransfer | Internal 🔒 | 🛑 | |
| L | safeTransferFrom | Internal 🔒 | 🛑 | |
| L | callOptionalReturn | Private 🔐 | 🛑 | |
| | | | | |
| **MainnetCDaiToCDaiAssimilator** | Implementation | | | |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | | NO ❗ |
| | | | | |
| **MainnetChaiToCDaiAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | viewNumeraireAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | | NO ❗ |
| | | | | |
| **MainnetDaiToCDaiAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | | NO ❗ |
| | | | | |
| **MainnetCUsdcToCUsdcAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | | NO ❗ |
| | | | | |
| **MainnetUsdcToCUsdcAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | | NO ❗ |
| | | | | |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| **MainnetCDaiToDaiAssimilator** | Implementation | | | |
| └ | | Public ❗ | 🛑 | NO ❗ |
| └ | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| └ | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| └ | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| └ | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| └ | outputRaw | Public ❗ | 🛑 | NO ❗ |
| └ | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| └ | viewRawAmount | Public ❗ | 🛑 | NO ❗ |
| └ | viewNumeraireAmount | Public ❗ | 🛑 | NO ❗ |
| └ | viewNumeraireAmountAndBalance | Public ❗ | 🛑 | NO ❗ |
| └ | viewNumeraireBalance | Public ❗ | 🛑 | NO ❗ |
| | | | | |
| **MainnetChaiToDaiAssimilator** | Implementation | | | |
| └ | | Public ❗ | 🛑 | NO ❗ |
| └ | add | Internal 🔒 | | |
| └ | sub | Internal 🔒 | | |
| └ | mul | Internal 🔒 | | |
| └ | rmul | Internal 🔒 | | |
| └ | rdivup | Internal 🔒 | | |
| └ | toDai | Internal 🔒 | | |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | fromDai | Internal 🔒 | | |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmountAndBalance | Public ❗ | 🛑 | NO ❗ |
| | | | | |
| **MainnetDaiToDaiAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | viewRawAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmountAndBalance | Public ❗ | 🛑 | NO ❗ |
| | | | | |
| **MainnetASUsdToSUsdAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | getASUsd | Public ❗ | | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmountAndBalance | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | 🛑 | NO ❗ |
| | | | | |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| **MainnetSUsdToSUsdAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmountAndBalance | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | 🛑 | NO ❗ |
| | | | | |
| **MainnetCUsdcToUsdcAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmountAndBalance | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | 🛑 | NO ❗ |
| | | | | |
| **MainnetUsdcToUsdcAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmountAndBalance | Public ❗ | 🛑 | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | viewNumeraireBalance | Public ❗ | 🛑 | NO ❗ |
| | | | | |
| **MainnetAUsdtToUsdtAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |
| L | getAUsdt | Private 🔐 | | |
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmountAndBalance | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | 🛑 | NO ❗ |
| L | safeTransfer | Internal 🔒 | 🛑 | |
| L | safeTransferFrom | Internal 🔒 | 🛑 | |
| L | callOptionalReturn | Private 🔐 | 🛑 | |
| | | | | |
| **MainnetUsdtToUsdtAssimilator** | Implementation | | | |
| L | | Public ❗ | 🛑 | NO ❗ |

| Contract | Type | Bases | | |
|---|---|---|---|---|
| L | intakeRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | intakeRaw | Public ❗ | 🛑 | NO ❗ |
| L | intakeNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | outputRawAndGetBalance | Public ❗ | 🛑 | NO ❗ |
| L | outputRaw | Public ❗ | 🛑 | NO ❗ |
| L | outputNumeraire | Public ❗ | 🛑 | NO ❗ |
| L | viewRawAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmount | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireAmountAndBalance | Public ❗ | 🛑 | NO ❗ |
| L | viewNumeraireBalance | Public ❗ | 🛑 | NO ❗ |
| L | safeTransfer | Internal 🔒 | 🛑 | |
| L | safeTransferFrom | Internal 🔒 | 🛑 | |
| L | callOptionalReturn | Private 🔐 | 🛑 | |

**Legend**

| Symbol | Meaning |
|---|---|
| 🛑 | Function can modify state |
| 💵 | Function is payable |

## A.2.4 Tests Suite

Below is the output generated by running the test suite:

Click to expand Test Suite Output

```
Running 13 tests for src/test/continuities/suiteSix.t.sol:ContinuitySuiteSix
[PASS] test_s6_continuity_synthesizedTargetswap_slippage() (gas: 794165)
[PASS] test_s6_continuity_synthesizedOriginSwap_slippage() (gas: 793406)
[PASS] test_s6_selectiveWithdraw_continuity_antiSlippage_reversal() (gas: 432626)
[PASS] test_s6_continuity_synthesizedTargetswap_antiSlippage() (gas: 793101)
[PASS] test_s6_selectiveDeposit_continuity_antiSlippage_reversal() (gas: 432245)
[FAIL] test_s6_selectiveDeposit_continuity_noSlippage_noAntiSlippage()
[PASS] test_s6_selectiveDeposit_continuity_slippage() (gas: 773548)
[PASS] test_s6_continuity_swap_slippage_reversals() (gas: 476628)
[PASS] test_s6_continuity_swap_antiSlippage_reversals() (gas: 473514)
[PASS] test_s6_selectiveDeposit_continuity_slippage_reversal() (gas: 432670)
[PASS] test_s6_selectiveDeposit_continuity_antiSlippage() (gas: 774397)
[PASS] test_s6_continuity_synthesizedOriginSwap_antiSlippage() (gas: 792352)
[PASS] test_s6_selectiveWithdraw_continuity_slippage_reversal() (gas: 432220)

Failure: test_s6_selectiveDeposit_continuity_noSlippage_noAntiSlippage

  "Error: Wrong `uint' value"
    Expected: 32500001
      Actual: 32500000


Running 2 tests for src/test/debug.t.sol:DebugTest
[PASS] testDebug() (gas: 2226)
[PASS] testMath() (gas: 360)

Running 20 tests for src/test/deposits/suiteFive.t.sol:SelectiveDepositSuiteFive
[PASS] test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_inBounds_
[OOPS] testFailSelectiveDepositUpperHaltCheck30Pct()
[PASS] test_s5_proportionalDeposit_monotonicity_lower_outOfBand() (gas: 518176)
[PASS] test_s5_proportionalDeposit_monotonicity_upper_outOfBand() (gas: 441497)
[OOPS] testFailSelectiveDepositDepostUpperHaltCheck10Pct()
[OOPS] test_s5_selectiveDeposit_monotonicity_upper_outOfBand_outOfBounds_to_outOfBour
[OOPS] test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_inBounds_
[OOPS] test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_outOfBour
[OOPS] test_s5_selectiveDeposit_monotonicity_upper_inBounds_to_outOfBounds_halt()
[OOPS] test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_outOfBour
[OOPS] test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_outOfBour
[OOPS] testFailSelectiveDepositLowerHaltCheck10Pct()
[OOPS] test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_inBounds_
[OOPS] test_s5_selectiveDeposit_monotonicity_upper_outOfBand_outOfBounds_to_outOfBour
[PASS] test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_inBounds_
[OOPS] test_s5_selectiveDeposit_monotonicity_upper_inBounds_to_outOfBounds_noHalt()
[OOPS] test_s5_selectiveDeposit_monotonicity_upper_outOfBand_outOfBounds_to_outOfBour
[OOPS] test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_outOfBour
[OOPS] test_s5_selectiveDeposit_monotonicity_upper_outOfBand_outOfBounds_to_outOfBour
```

```
[OOPS] testFailSelectiveDepositLowerHaltCheck30Pct()

VM error for testFailSelectiveDepositUpperHaltCheck30Pct()
VM error for testFailSelectiveDepositDepostUpperHaltCheck10Pct()
VM error for test_s5_selectiveDeposit_monotonicity_upper_outOfBand_outOfBounds_to_out
VM error for test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_inB
VM error for test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_out
VM error for test_s5_selectiveDeposit_monotonicity_upper_inBounds_to_outOfBounds_halt
VM error for test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_out
VM error for test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_out
VM error for testFailSelectiveDepositLowerHaltCheck10Pct()
VM error for test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_inB
VM error for test_s5_selectiveDeposit_monotonicity_upper_outOfBand_outOfBounds_to_out
VM error for test_s5_selectiveDeposit_monotonicity_upper_inBounds_to_outOfBounds_noHa
VM error for test_s5_selectiveDeposit_monotonicity_upper_outOfBand_outOfBounds_to_out
VM error for test_s5_selectiveDeposit_monotonicity_lower_outOfBand_outOfBounds_to_out
VM error for test_s5_selectiveDeposit_monotonicity_upper_outOfBand_outOfBounds_to_out
VM error for testFailSelectiveDepositLowerHaltCheck30Pct()
Running 37 tests for src/test/deposits/suiteOne.t.sol:SelectiveDepositSuiteOne
[OOPS] testFailSelectiveDepositUpperHaltCheck30Pct()
[PASS] test_s1_selectiveDeposit_noSlippage_36DAI_from_300Proportional() (gas: 329705)
[PASS] test_s1_selectiveDeposit_smartHalt_lower_outOfBounds_to_inBounds() (gas: 45126
[PASS] test_s1_selectiveDeposit_balanced_5DAI_1USDC_3USDT_1SUSD() (gas: 387605)
[PASS] test_s1_selectiveDeposit_fullUpperSlippage_5USDC_3SUSD_into_90DAI_145USDC_90US
[PASS] test_s1_selectiveDeposit_partialUpperAntiSlippage_unbalanced_1DAI_46USDC_53USD
[PASS] test_s1_selectiveDeposit_megaDepositDirectLowerToUpper_105DAI_37SUSD_from_55DA
[FAIL] test_s1_selectiveDeposit_megaDepositIndirectUpperLower_165CDAI_0p0001CUSDC_165
[PASS] test_s1_selectiveDeposit_fullUpperAntiSlippage_8DAI_12USDC_10USDT_2SUSD_into_1
[OOPS] testExecuteProportionalDepositIntoWidelyUnbalancedShell()
[PASS] test_s1_selectiveDeposit_fullLowerSlippage_12DAI_12USDC_1USDT_1SUSD_into_95DAI
[PASS] test_s1_selectiveDeposit_partialUpperAntiSlippage_46USDC_53USDT_into_145DAI_90
[PASS] test_s1_selectiveDeposit_fullLowerAntiSlippage_5DAI_5USDC_5USDT_2SUSD_into_55D
[OOPS] testFailSelectiveDepositDepostUpperHaltCheck10Pct()
[PASS] test_s1_selectiveDeposit_fullLowerSlippage_9DAI_9USDC_into_95DAI_95USDC_55USDT
[PASS] test_s1_selectiveDeposit_partialLowerSlippage_95DAI_55USDC_95USDT_15SUSD() (ga
[FAIL] test_s1_selectiveDeposit_smartHalt_lower_outOfBounds_to_outOfBounds()
[PASS] test_s1_selectiveDeposit_smartHalt_lower_unrelated() (gas: 458526)
[PASS] test_s1_selectiveDeposit_smartHalt_upper_outOfBounds_to_inBounds() (gas: 40584
[PASS] test_s1_selectiveDeposit_partialLowerSlippage_moderatelyUnbalanced_1DAI_51USDC
[PASS] test_s1_selectiveDeposit_noSlippage_36CHAI_into_300Proportional() (gas: 343864
[PASS] test_s1_selectiveDeposit_smartHalt_upper_outOfBounds_to_outOfBounds() (gas: 40
[PASS] test_s1_selectiveDeposit_partialLowerAntiSlippage_36CUSDC_18ASUSD_into_95DAI_5
[PASS] test_s1_selectiveDeposit_partialUpperSlippage_5DAI_5USDC_70USDT_28SUSD_300Prop
[PASS] test_s1_selectiveDeposit_partialUpperSlippage_145DAI_90USDC_90USDT_50SUSD() (g
[OOPS] testFailSelectiveDepositLowerHaltCheck10Pct()
[PASS] test_s1_selectiveDeposit_upperSlippage_36Point001Dai_into_300Proportional() (g
[PASS] test_s1_selectiveDeposit_fullUpperAntiSlippage_5CHAI_5USDC_into_90DAI_90USDC_1
[PASS] test_s1_selectiveDeposit_partialLowerAntiSlippage_36USDC_18SUSD_into_95DAI_55U
[PASS] test_s1_selectiveDeposit_smartHalt_upper_outOfBounds_exacerbated() (gas: 35055
[OOPS] testExecuteProportionalDepositIntoAnUnbalancedShell()
[PASS] test_s1_selectiveDeposit_noSlippage_36CDAI_into_300Proportional() (gas: 375384
[PASS] test_s1_selectiveDeposit_partialLowerSlippage_balanced_0p001DAI_90USDC_90USDT(
```

```
[OOPS] testExecuteProportionalDepositIntoSlightlyUnbalancedShell()
[PASS] test_s1_selectiveDeposit_megaDepositIndirectUpperToLower_165DAI_165USDT_into_9
[PASS] test_s1_selectiveDeposit_megaDepositIndirectUpperToLower_165DAI_0p0001USDC_165
[OOPS] testFailSelectiveDepositLowerHaltCheck30Pct()

VM error for testFailSelectiveDepositUpperHaltCheck30Pct()
Failure: test_s1_selectiveDeposit_megaDepositIndirectUpperLower_165CDAI_0p0001CUSDC_1

  "Error: Wrong `uint' value"
    Expected: 32007966147966147958
      Actual: 334941012602709411694


VM error for testExecuteProportionalDepositIntoWidelyUnbalancedShell()
VM error for testFailSelectiveDepositDepostUpperHaltCheck10Pct()
Failure: test_s1_selectiveDeposit_smartHalt_lower_outOfBounds_to_outOfBounds

  "Assertion failed"


VM error for testFailSelectiveDepositLowerHaltCheck10Pct()
VM error for testExecuteProportionalDepositIntoAnUnbalancedShell()
VM error for testExecuteProportionalDepositIntoSlightlyUnbalancedShell()
VM error for testFailSelectiveDepositLowerHaltCheck30Pct()
Running 7 tests for src/test/deposits/suiteSix.t.sol:SelectiveDepositSuiteSix
[OOPS] testFailSelectiveDepositUpperHaltCheck30Pct()
[OOPS] testFailSelectiveDepositDepostUpperHaltCheck10Pct()
[PASS] test_s6_selectiveDeposit_continuity_noSlippage_noAntiSlippage() (gas: 524188)
[PASS] test_s6_selectiveDeposit_continuity_slippage() (gas: 773534)
[OOPS] testFailSelectiveDepositLowerHaltCheck10Pct()
[PASS] test_s6_selectiveDeposit_continuity_antiSlippage() (gas: 774319)
[OOPS] testFailSelectiveDepositLowerHaltCheck30Pct()

VM error for testFailSelectiveDepositUpperHaltCheck30Pct()
VM error for testFailSelectiveDepositDepostUpperHaltCheck10Pct()
VM error for testFailSelectiveDepositLowerHaltCheck10Pct()
VM error for testFailSelectiveDepositLowerHaltCheck30Pct()
Running 37 tests for src/test/deposits/suiteTwo.t.sol:SelectiveDepositSuiteTwo
[OOPS] testFailSelectiveDepositUpperHaltCheck30Pct()
[FAIL] test_s2_selectiveDeposit_noSlippage_balanced_10DAI_10USDC_10USDT_2p5SUSD()
[FAIL] test_s2_selectiveDeposit_fullUpperSlippage_5USDC_3SUSD_into_90DAI_145USDC_90US
[FAIL] test_s2_selectiveDeposit_fullUpperAntiSlippage_8DAI_12USDC_10USDT_2SUSD_into_1
[FAIL] test_s2_selectiveDeposit_partialLowerAntiSlippage_36USDC_18SUSD_into_95DAI_55U
[FAIL] test_s2_selectiveDeposit_fullLowerSlippage_12DAI_12USDC_1USDT_1SUSD_into_95DAI
[OOPS] testExecuteProportionalDepositIntoWidelyUnbalancedShell()
[FAIL] test_s2_selectiveDeposit_fullUpperAntiSlippage_5DAI_5USDC_into_90DAI_90USDC_14
[OOPS] testFailSelectiveDepositDepostUpperHaltCheck10Pct()
[FAIL] test_s2_selectiveDeposit_megaDepositIndirectUpperToLower_165DAI_0p0001USDC_165
[FAIL] test_s2_selectiveDeposit_megaDepositDirectLowerToUpper_105DAI_37SUSD_from_55DA
[FAIL] test_s2_selectiveDeposit_partialLowerSlippage_balanced_0p001DAI_90USDC_90USDT(
[FAIL] test_s2_selectiveDeposit_upperSlippage_36Point001Dai_into_300Proportional()
[FAIL] test_s2_selectiveDeposit_partialLowerAntiSlippage_36CUSDC_18ASUSD_into_95DAI_5
```

```
[FAIL] test_s2_selectiveDeposit_partialLowerSlippage_moderatelyUnbalanced_1DAI_51USDC
[FAIL] test_s2_selectiveDeposit_noSlippage_36CDAI_into_300Proportional()
[FAIL] test_s2_selectiveDeposit_partialUpperAntiSlippage_46USDC_53USDT_into_145DAI_90
[FAIL] test_s2_selectiveDeposit_balanced_5DAI_1USDC_3USDT_1SUSD()
[PASS] test_s2_selectiveDeposit_smartHalt_upper_outOfBounds_to_inBounds() (gas: 40591
[FAIL] test_s2_selectiveDeposit_fullLowerAntiSlippage_5DAI_5USDC_5USDT_2SUSD_into_55D
[FAIL] test_s2_selectiveDeposit_noSlippage_36DAI_from_300Proportional()
[OOPS] testFailSelectiveDepositLowerHaltCheck10Pct()
[FAIL] test_s2_selectiveDeposit_megaDepositIndirectUpperLower_165CDAI_0p0001CUSDC_165
[PASS] test_s2_selectiveDeposit_smartHalt_upper_outOfBounds_to_outOfBounds() (gas: 40
[FAIL] test_s2_selectiveDeposit_partialUpperAntiSlippage_unbalanced_1DAI_46USDC_53USD
[OOPS] testExecuteProportionalDepositIntoAnUnbalancedShell()
[OOPS] testExecuteProportionalDepositIntoSlightlyUnbalancedShell()
[FAIL] test_s2_selectiveDeposit_noSlippage_36CHAI_into_300Proportional()
[FAIL] test_s2_selectiveDeposit_partialUpperSlippage_5DAI_5USDC_70USDT_28SUSD_300Prop
[FAIL] test_s2_selectiveDeposit_megaDepositIndirectUpperToLower_165DAI_165USDT_into_9
[FAIL] test_s2_selectiveDeposit_fullUpperAntiSlippage_5CHAI_5USDC_into_90DAI_90USDC_1
[FAIL] test_s2_selectiveDeposit_fullLowerSlippage_9DAI_9USDC_into_95DAI_95USDC_55USDT
[FAIL] test_s2_selectiveDeposit_partialUpperSlippage_145DAI_90USDC_90USDT_50SUSD()
[FAIL] test_s2_selectiveDeposit_smartHalt_lower_outOfBounds_to_outOfBounds()
[PASS] test_s2_selectiveDeposit_smartHalt_lower_outOfBounds_to_inBounds() (gas: 45128
[OOPS] testFailSelectiveDepositLowerHaltCheck30Pct()
[FAIL] test_s2_selectiveDeposit_partialLowerSlippage_95DAI_55USDC_95USDT_15SUSD()

VM error for testFailSelectiveDepositUpperHaltCheck30Pct()
Failure: test_s2_selectiveDeposit_noSlippage_balanced_10DAI_10USDC_10USDT_2p5SUSD

  "Error: Wrong `uint' value"
    Expected: 32499999216641686631
      Actual: 32499999999999999988


Failure: test_s2_selectiveDeposit_fullUpperSlippage_5USDC_3SUSD_into_90DAI_145USDC_90

  "Error: Wrong `uint' value"
    Expected: 7939105448732499106
      Actual: 7939106469393675653


Failure: test_s2_selectiveDeposit_fullUpperAntiSlippage_8DAI_12USDC_10USDT_2SUSD_into

  "Error: Wrong `uint' value"
    Expected: 32007965048728686700
      Actual: 32007966147966147958


Failure: test_s2_selectiveDeposit_partialLowerAntiSlippage_36USDC_18SUSD_into_95DAI_5

  "Error: Wrong `uint' value"
    Expected: 54018716739832990695
      Actual: 54018717948717948711
```

```
Failure: test_s2_selectiveDeposit_fullLowerSlippage_12DAI_12USDC_1USDT_1SUSD_into_95D

  "Error: Wrong `uint' value"
    Expected: 25908472086895042433
      Actual: 25908473193473193467


VM error for testExecuteProportionalDepositIntoWidelyUnbalancedShell()
Failure: test_s2_selectiveDeposit_fullUpperAntiSlippage_5DAI_5USDC_into_90DAI_90USDC_

  "Error: Wrong `uint' value"
    Expected: 10006716145229473334
      Actual: 10006717171717171714


VM error for testFailSelectiveDepositDepostUpperHaltCheck10Pct()
Failure: test_s2_selectiveDeposit_megaDepositIndirectUpperToLower_165DAI_0p0001USDC_1

  "Error: Wrong `uint' value"
    Expected: 330445739346952556280
      Actual: 330445741274888467979


Failure: test_s2_selectiveDeposit_megaDepositDirectLowerToUpper_105DAI_37SUSD_from_55

  "Error: Wrong `uint' value"
    Expected: 142003004834841080526
      Actual: 142003004847557086355


Failure: test_s2_selectiveDeposit_partialLowerSlippage_balanced_0p001DAI_90USDC_90USD

  "Error: Wrong `uint' value"
    Expected: 179701018321068682614
      Actual: 179701018124533421095


Failure: test_s2_selectiveDeposit_upperSlippage_36Point001Dai_into_300Proportional

  "Error: Wrong `uint' value"
    Expected: 36001000238070333757
      Actual: 36000999999612476342


Failure: test_s2_selectiveDeposit_partialLowerAntiSlippage_36CUSDC_18ASUSD_into_95DAI

  "Error: Wrong `uint' value"
    Expected: 53991711756245652892
      Actual: 54018716948717948714
```

```
Failure: test_s2_selectiveDeposit_partialLowerSlippage_moderatelyUnbalanced_1DAI_51US

  "Error: Wrong `uint' value"
    Expected: 103803800870238866890
      Actual: 103803802211302211279


Failure: test_s2_selectiveDeposit_noSlippage_36CDAI_into_300Proportional

  "Error: Wrong `uint' value"
    Expected: 35991000239800010000
      Actual: 35999999999852135533


Failure: test_s2_selectiveDeposit_partialUpperAntiSlippage_46USDC_53USDT_into_145DAI_

  "Error: Wrong `uint' value"
    Expected: 99008609844270035541
      Actual: 99008611111111111104


Failure: test_s2_selectiveDeposit_balanced_5DAI_1USDC_3USDT_1SUSD

  "Error: Wrong `uint' value"
    Expected: 9999998966167174500
      Actual: 9999999999999999991


Failure: test_s2_selectiveDeposit_fullLowerAntiSlippage_5DAI_5USDC_5USDT_2SUSD_into_5

  "Error: Wrong `uint' value"
    Expected: 17007126629845201617
      Actual: 17007127696010367489


Failure: test_s2_selectiveDeposit_noSlippage_36DAI_from_300Proportional

  "Error: Wrong `uint' value"
    Expected: 36000000233425481370
      Actual: 35999999999999999985


VM error for testFailSelectiveDepositLowerHaltCheck10Pct()
Failure: test_s2_selectiveDeposit_megaDepositIndirectUpperLower_165CDAI_0p0001CUSDC_1

  "Error: Wrong `uint' value"
    Expected: 33028053905716828894
      Actual: 33494101260270941169


Failure: test_s2_selectiveDeposit_partialUpperAntiSlippage_unbalanced_1DAI_46USDC_53U
```

```
  "Error: Wrong `uint' value"
    Expected: 101008609838582174525
      Actual: 101008611111111111102


VM error for testExecuteProportionalDepositIntoAnUnbalancedShell()
VM error for testExecuteProportionalDepositIntoSlightlyUnbalancedShell()
Failure: test_s2_selectiveDeposit_noSlippage_36CHAI_into_300Proportional

  "Error: Wrong `uint' value"
    Expected: 35991000233367100000
      Actual: 35999999999999999985


Failure: test_s2_selectiveDeposit_partialUpperSlippage_5DAI_5USDC_70USDT_28SUSD_300Pr

  "Error: Wrong `uint' value"
    Expected: 107839868987150692242
      Actual: 107839869281045751654


Failure: test_s2_selectiveDeposit_megaDepositIndirectUpperToLower_165DAI_165USDT_into

  "Error: Wrong `uint' value"
    Expected: 329943557873174181881
      Actual: 329943557919621749370


Failure: test_s2_selectiveDeposit_fullUpperAntiSlippage_5CHAI_5USDC_into_90DAI_90USDC

  "Error: Wrong `uint' value"
    Expected: 10001714411049177790
      Actual: 10006716171387577028


Failure: test_s2_selectiveDeposit_fullLowerSlippage_9DAI_9USDC_into_95DAI_95USDC_55US

  "Error: Wrong `uint' value"
    Expected: 17902137819144617096
      Actual: 17902138904261206411


Failure: test_s2_selectiveDeposit_partialUpperSlippage_145DAI_90USDC_90USDT_50SUSD

  "Error: Wrong `uint' value"
    Expected: 374956943424882834388
      Actual: 374956944444444444455


Failure: test_s2_selectiveDeposit_smartHalt_lower_outOfBounds_to_outOfBounds

  "Assertion failed"
```

```
VM error for testFailSelectiveDepositLowerHaltCheck30Pct()
Failure: test_s2_selectiveDeposit_partialLowerSlippage_95DAI_55USDC_95USDT_15SUSD

  "Error: Wrong `uint' value"
    Expected: 259906409242241292207
      Actual: 259906410256410256403


Running 37 tests for src/test/deposits/views/suiteOneViews.t.sol:SelectiveDepositSuit
[OOPS] testFailSelectiveDepositUpperHaltCheck30Pct()
[PASS] test_s1_selectiveDepositViews_smartHalt_upper_outOfBounds_exacerbated() (gas:
[PASS] test_s1_selectiveDepositViews_smartHalt_upper_outOfBounds_to_outOfBounds() (ga
[PASS] test_s1_selectiveDepositViews_fullLowerSlippage_12DAI_12USDC_1USDT_1SUSD_into_
[PASS] test_s1_selectiveDepositViews_smartHalt_lower_outOfBounds_to_inBounds() (gas:
[PASS] test_s1_selectiveDepositViews_fullLowerSlippage_9DAI_9USDC_into_95DAI_95USDC_5
[OOPS] testExecuteProportionalDepositIntoWidelyUnbalancedShell()
[PASS] test_s1_selectiveDepositViews_smartHalt_lower_unrelated() (gas: 458505)
[FAIL] test_s1_selectiveDepositViews_noSlippage_36CDAI_into_300Proportional()
[OOPS] testFailSelectiveDepositDepostUpperHaltCheck10Pct()
[PASS] test_s1_selectiveDepositViews_smartHalt_upper_outOfBounds_to_inBounds() (gas:
[FAIL] test_s1_selectiveDepositViews_smartHalt_lower_outOfBounds_to_outOfBounds()
[PASS] test_s1_selectiveDepositViews_partialUpperAntiSlippage_unbalanced_1DAI_46USDC_
[PASS] test_s1_selectiveDepositViews_megaDepositIndirectUpperToLower_165DAI_165USDT_i
[FAIL] test_s1_selectiveDepositViews_noSlippage_36CHAI_into_300Proportional()
[PASS] test_s1_selectiveDepositViews_megaDepositDirectLowerToUpper_105DAI_37SUSD_from
[PASS] test_s1_selectiveDepositViews_partialLowerSlippage_balanced_0p001DAI_90USDC_90
[PASS] test_s1_selectiveDepositViews_fullUpperAntiSlippage_5CHAI_5USDC_into_90DAI_90U
[PASS] test_s1_selectiveDepositViews_partialLowerSlippage_moderatelyUnbalanced_1DAI_5
[PASS] test_s1_selectiveDepositViews_partialLowerSlippage_95DAI_55USDC_95USDT_15SUSD(
[PASS] test_s1_selectiveDepositViews_partialUpperSlippage_145DAI_90USDC_90USDT_50SUSD
[PASS] test_s1_selectiveDepositViews_fullUpperAntiSlippage_8DAI_12USDC_10USDT_2SUSD_i
[FAIL] test_s1_selectiveDepositViews_partialLowerAntiSlippage_36CUSDC_18ASUSD_into_95
[PASS] test_s1_selectiveDepositViews_noSlippage_36DAI_from_300Proportional() (gas: 30
[OOPS] testFailSelectiveDepositLowerHaltCheck10Pct()
[PASS] test_s1_selectiveDepositViews_megaDepositIndirectUpperToLower_165DAI_0p0001USD
[PASS] test_s1_selectiveDepositViews_partialLowerAntiSlippage_36USDC_18SUSD_into_95DA
[PASS] test_s1_selectiveDepositViews_balanced_5DAI_1USDC_3USDT_1SUSD() (gas: 324078)
[OOPS] testExecuteProportionalDepositIntoAnUnbalancedShell()
[FAIL] test_s1_selectiveDepositViews_megaDepositIndirectUpperLower_165CDAI_0p0001CUSD
[PASS] test_s1_selectiveDepositViews_partialUpperSlippage_5DAI_5USDC_70USDT_28SUSD_30
[OOPS] testExecuteProportionalDepositIntoSlightlyUnbalancedShell()
[PASS] test_s1_selectiveDepositViews_partialUpperAntiSlippage_46USDC_53USDT_into_145D
[PASS] test_s1_selectiveDepositViews_fullUpperSlippage_5USDC_3SUSD_into_90DAI_145USDC
[PASS] test_s1_selectiveDepositViews_fullLowerAntiSlippage_5DAI_5USDC_5USDT_2SUSD_int
[PASS] test_s1_selectiveDepositViews_upperSlippage_36Point001Dai_into_300Proportional
[OOPS] testFailSelectiveDepositLowerHaltCheck30Pct()

VM error for testFailSelectiveDepositUpperHaltCheck30Pct()
VM error for testExecuteProportionalDepositIntoWidelyUnbalancedShell()
Failure: test_s1_selectiveDepositViews_noSlippage_36CDAI_into_300Proportional
```

```
  "Error: Wrong `uint' value"
    Expected: 35991000239800010000
      Actual: 35999999999852135533


VM error for testFailSelectiveDepositDepostUpperHaltCheck10Pct()
Failure: test_s1_selectiveDepositViews_smartHalt_lower_outOfBounds_to_outOfBounds

  "Assertion failed"


Failure: test_s1_selectiveDepositViews_noSlippage_36CHAI_into_300Proportional

  "Error: Wrong `uint' value"
    Expected: 35991000233367100000
      Actual: 35999999999999999985


Failure: test_s1_selectiveDepositViews_partialLowerAntiSlippage_36CUSDC_18ASUSD_into_

  "Error: Wrong `uint' value"
    Expected: 53991711756245652892
      Actual: 54018716948717948714


VM error for testFailSelectiveDepositLowerHaltCheck10Pct()
VM error for testExecuteProportionalDepositIntoAnUnbalancedShell()
Failure: test_s1_selectiveDepositViews_megaDepositIndirectUpperLower_165CDAI_0p0001CU

  "Error: Wrong `uint' value"
    Expected: 32007966147966147958
      Actual: 33494101260270941169 4


VM error for testExecuteProportionalDepositIntoSlightlyUnbalancedShell()
VM error for testFailSelectiveDepositLowerHaltCheck30Pct()
Running 7 tests for src/test/originSwaps/suiteFive.t.sol:OriginSwapSuiteFiveTest
[OOPS] test_s5_originSwap_monotonicity_outOfBand_mutuallyOutOfBound_towards_mutuallyI
[OOPS] test_s5_originSwap_monotonicity_mutuallyInBounds_to_mutuallyOutOfBounds_halts(
[PASS] test_s5_originSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuallyInBou
[OOPS] test_s5_originSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuallyOutOf
[OOPS] test_s5_originSwap_monotonicity_mutuallyInBounds_to_mutuallyOutOfBounds_noHalt
[OOPS] test_s5_originSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuallyOutOf
[OOPS] test_s5_originSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuallyInBou

VM error for test_s5_originSwap_monotonicity_outOfBand_mutuallyOutOfBound_towards_mut
VM error for test_s5_originSwap_monotonicity_mutuallyInBounds_to_mutuallyOutOfBounds_
VM error for test_s5_originSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuall
VM error for test_s5_originSwap_monotonicity_mutuallyInBounds_to_mutuallyOutOfBounds_
VM error for test_s5_originSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuall
VM error for test_s5_originSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuall
Running 34 tests for src/test/originSwaps/suiteOne.t.sol:OriginSwapSuiteOneTest
```

```
Running 34 tests for src/test/originSwaps/suiteOne.t.sol:OriginSwapSuiteOneTest
[PASS] test_s1_originSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_CUSDC_to_
[PASS] test_s1_originSwap_upperHaltCheck_10PctWeight() (gas: 330050)
[PASS] test_s1_originSwap_smartHalt_lower_outOfBounds_to_outOfBounds() (gas: 476128)
[PASS] test_s1_originSwap_megaLowerToUpperUpperToLower_CDAI_30PctWeight() (gas: 39198
[PASS] test_s1_originSwap_lowerhaltCheck_10PctWeight() (gas: 340486)
[PASS] test_s1_originSwap_megaLowerToUpperUpperToLower_30PctWeight() (gas: 346262)
[PASS] test_s1_originSwap_megaUpperToLower_30PctWeight_to_10PctWeight() (gas: 345442)
[PASS] test_s1_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight_to_
[OOPS] testFailOriginSwap_greaterThanBalance_10Pct()
[PASS] test_s1_originSwap_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct() (gas:
[PASS] test_s1_originSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight_to_10PctWe
[OOPS] testFailOriginSwap_greaterThanBalance_30Pct()
[PASS] test_s1_originSwap_CHAI_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct() (
[PASS] test_s1_originSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight__() (gas:
[PASS] test_s1_originSwap_fullUpperAndLowerAntiSlippage_unbalanced_30PctWeight() (gas
[PASS] test_s1_originSwap_partialUpperAndLowerSlippage_unbalanced_10PctWeight_to_30Pc
[PASS] test_s1_originSwap_smartHalt_upper() (gas: 384568)
[PASS] test_s1_originSwap_noSlippage_balanced_30PctWeight_to_30PctWeight() (gas: 3331
[PASS] test_s1_originSwap_smartHalt_lower_outOfBounds_to_inBounds() (gas: 468986)
[PASS] test_s1_originSwap_smartHalt_upper_unrelated() (gas: 404118)
[PASS] test_s1_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight__()
[PASS] test_s1_originSwap_lowerHaltCheck_30PctWeight() (gas: 354223)
[PASS] test_s1_originSwap_megaLowerToUpper_10PctWeight_to_30PctWeight() (gas: 345352)
[PASS] test_s1_originSwap_noSlippage_lightlyUnbalanced_10USDC_to_USDT_with_80DAI_100U
[PASS] test_s1_originSwap_noSlippage_balanced_10DAI_to_USDC_300Proportional() (gas: 3
[PASS] test_s1_originSwap_fullUpperAndLowerAntiSlippage_10PctWeight_to30PctWeight() (
[PASS] test_s1_originSwap_smartHalt_lower_unrelated() (gas: 404117)
[PASS] test_s1_originSwap_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight() (
[PASS] test_s1_originSwap_fullUpperAndLowerSlippage_unbalanced_10PctWeight_to_30PctWe
[PASS] test_s1_originSwap_upperHaltCheck_30PctWeight() (gas: 351011)
[PASS] test_s1_originSwap_partialUpperAndLowerSlippage_balanced_40USDC_to_DAI() (gas:
[PASS] test_s1_originSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_to_10PctW
[PASS] test_s1_originSwap_noSlippage_balanced_10PctWeight_to_30PctWeight() (gas: 3329
[PASS] test_s1_originSwap_fullUpperAndLowerSlippage_CUSDC_ASUSD_unbalanced_10PctWeigh

VM error for testFailOriginSwap_greaterThanBalance_10Pct()
VM error for testFailOriginSwap_greaterThanBalance_30Pct()
Running 35 tests for src/test/originSwaps/suiteSeven.t.sol:OriginSwapSuiteOneTest
[FAIL] test_s7_originSwap_fullUpperAndLowerAntiSlippage_10PctWeight_to30PctWeight()
[FAIL] test_s7_originSwap_CHAI_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct()
[FAIL] test_s7_originSwap_megaUpperToLower_30PctWeight_to_10PctWeight()
[FAIL] test_s7_originSwap_noSlippage_balanced_30PctWeight_to_30PctWeight()
[PASS] test_s7_originSwap_smartHalt_lower_outOfBounds_to_inBounds() (gas: 468943)
[PASS] test_s7_originSwap_smartHalt_lower_outOfBounds_to_outOfBounds() (gas: 476105)
[FAIL] test_s7_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_10PctWeight_to_
[FAIL] test_s7_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight_to_
[OOPS] testFailOriginSwap_greaterThanBalance_10Pct()
[FAIL] test_s7_originSwap_partialUpperAndLowerSlippage_balanced_40USDC_to_DAI()
[FAIL] test_s7_originSwap_fullUpperAndLowerSlippage_CUSDC_ASUSD_unbalanced_10PctWeigh
[FAIL] test_s7_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight__()
[OOPS] testFailOriginSwap_greaterThanBalance_30Pct()
[PASS] test_s7_originSwap_smartHalt_upper_unrelated() (gas: 404184)
```

```
[PASS] test_s7_originSwap_smartHalt_upper_unrelated() (gas: [illegible])
[PASS] test_s7_originSwap_smartHalt_lower_unrelated() (gas: 404161)
[PASS] test_s7_originSwap_upperHaltCheck_10PctWeight() (gas: 330026)
[FAIL] test_s7_originSwap_noSlippage_balanced_10DAI_to_USDC_300Proportional()
[FAIL] test_s7_originSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_CUSDC_to_
[FAIL] test_s7_originSwap_megaLowerToUpper_10PctWeight_to_30PctWeight()
[FAIL] test_s7_originSwap_megaLowerToUpperUpperToLower_CDAI_30PctWeight()
[PASS] test_s7_originSwap_upperHaltCheck_30PctWeight() (gas: 351055)
[FAIL] test_s7_originSwap_fullUpperAndLowerAntiSlippage_unbalanced_30PctWeight()
[FAIL] test_s7_originSwap_megaLowerToUpperUpperToLower_30PctWeight()
[FAIL] test_s7_originSwap_noSlippage_balanced_10PctWeight_to_30PctWeight()
[FAIL] test_s7_originSwap_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct()
[FAIL] test_s7_originSwap_partialUpperAndLowerSlippage_unbalanced_10PctWeight_to_30Pc
[FAIL] test_s7_originSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight()
[FAIL] test_s7_originSwap_noSlippage_lightlyUnbalanced_10USDC_to_USDT_with_80DAI_100U
[PASS] test_s7_originSwap_lowerHaltCheck_30PctWeight() (gas: 354265)
[PASS] test_s7_originSwap_smartHalt_upper() (gas: 384634)
[FAIL] test_s7_originSwap_fullUpperAndLowerSlippage_unbalanced_10PctWeight_to_30PctWe
[FAIL] test_s7_originSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight_to_10PctWe
[FAIL] test_s7_originSwap_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight()
[FAIL] test_s7_originSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_to_10PctW
[PASS] test_s7_originSwap_lowerhaltCheck_10PctWeight() (gas: 340527)

Failure: test_s7_originSwap_fullUpperAndLowerAntiSlippage_10PctWeight_to30PctWeight

  "Error: Wrong `uint' value"
    Expected: 3660153
      Actual: 3661067


Failure: test_s7_originSwap_CHAI_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct

  "Error: Wrong `uint' value"
    Expected: 2365464484251272960
      Actual: 2366053853162344119


Failure: test_s7_originSwap_megaUpperToLower_30PctWeight_to_10PctWeight

  "Error: Wrong `uint' value"
    Expected: 19990016481618381864
      Actual: 19994999999999999972


Failure: test_s7_originSwap_noSlippage_balanced_30PctWeight_to_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 9995000
      Actual: 9997499


Failure: test_s7_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_10PctWeight_t
```

```
  "Error: Wrong `uint' value"
    Expected: 10006174300378984359
      Actual: 10008673676470588224


Failure: test_s7_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight_t

  "Error: Wrong `uint' value"
    Expected: 10019788191004510065
      Actual: 10022287566546762578


VM error for testFailOriginSwap_greaterThanBalance_10Pct()
Failure: test_s7_originSwap_partialUpperAndLowerSlippage_balanced_40USDC_to_DAI

  "Error: Wrong `uint' value"
    Expected: 39330195827959985796
      Actual: 39339756348795716299


Failure: test_s7_originSwap_fullUpperAndLowerSlippage_CUSDC_ASUSD_unbalanced_10PctWei

  "Error: Wrong `uint' value"
    Expected: 2696349000000000000
      Actual: 2697033999999999999


Failure: test_s7_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight__

  "Error: Wrong `uint' value"
    Expected: 30070278169642344458
      Actual: 30077776294642857112


VM error for testFailOriginSwap_greaterThanBalance_30Pct()
Failure: test_s7_originSwap_noSlippage_balanced_10DAI_to_USDC_300Proportional

  "Error: Wrong `uint' value"
    Expected: 9995000
      Actual: 9997499


Failure: test_s7_originSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_CUSDC_t

  "Error: Wrong `uint' value"
    Expected: 39330195827959985796
      Actual: 39339755421580561359


Failure: test_s7_originSwap_megaLowerToUpper_10PctWeight_to_30PctWeight

  "Error: Wrong `uint' value"
```

```
    Expected: 19990003
      Actual: 19994999


Failure: test_s7_originSwap_megaLowerToUpperUpperToLower_CDAI_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 17491279
      Actual: 69982499


Failure: test_s7_originSwap_fullUpperAndLowerAntiSlippage_unbalanced_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 5045804
      Actual: 5047059


Failure: test_s7_originSwap_megaLowerToUpperUpperToLower_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 69965119
      Actual: 69982499


Failure: test_s7_originSwap_noSlippage_balanced_10PctWeight_to_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 3998000
      Actual: 3998999


Failure: test_s7_originSwap_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct

  "Error: Wrong `uint' value"
    Expected: 2365464484251272960
      Actual: 2366053853162344119


Failure: test_s7_originSwap_partialUpperAndLowerSlippage_unbalanced_10PctWeight_to_30

  "Error: Wrong `uint' value"
    Expected: 7920411672881948283
      Actual: 7922386282489836276


Failure: test_s7_originSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 4666173
      Actual: 4667368
```

```
Failure: test_s7_originSwap_noSlippage_lightlyUnbalanced_10USDC_to_USDT_with_80DAI_10

  "Error: Wrong `uint' value"
    Expected: 9995000
      Actual: 9997499


Failure: test_s7_originSwap_fullUpperAndLowerSlippage_unbalanced_10PctWeight_to_30Pct

  "Error: Wrong `uint' value"
    Expected: 2696349
      Actual: 2697035


Failure: test_s7_originSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight_to_10Pct

  "Error: Wrong `uint' value"
    Expected: 2876384124908864750
      Actual: 2877116893342516205


Failure: test_s7_originSwap_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight

  "Error: Wrong `uint' value"
    Expected: 2998500187500000000
      Actual: 2999249999999999997


Failure: test_s7_originSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_to_10Pc

  "Error: Wrong `uint' value"
    Expected: 14813513177462324025
      Actual: 14817098101815228528


Running 8 tests for src/test/originSwaps/suiteSix.t.sol:OriginSwapSuiteSixTest
[PASS] test_s6_originSwap_continuity_partialUpperAndLowerAntiSlippage_unbalanced_10Pc
[PASS] test_s6_originSwap_continuity_30Pct_to_30Pct() (gas: 835043)
[PASS] test_s6_originSwap_continuity_fullUpperAndLowerAntiSlippage_30Pct_to_10Pct() (
[PASS] test_s6_originSwap_continuity_partialUpperAndLowerFees_30Pct_to_10Pct() (gas:
[PASS] test_s6_originSwap_continuity_upperAndLowerFees_30Pct_to_30Pct() (gas: 800669)
[PASS] test_s6_originSwap_continuity_partialUpperAndLowerAntiSlippage_unbalanced_30Pc
[PASS] test_s6_originSwap_continuity_partialUpperAndLowerAntiSlippage_30Pct_to_30Pct(
[PASS] test_s6_originSwap_continuity_fullUpperAndLowerAntiSlippage_30Pct_to_30Pct() (

Running 35 tests for src/test/originSwaps/suiteTwo.t.sol:OriginSwapSuiteOneTest
[FAIL] test_s2_originSwap_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct()
[FAIL] test_s2_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight_to_
[PASS] test_s2_originSwap_lowerHaltCheck_30PctWeight() (gas: 354245)
[PASS] test_s2_originSwap_smartHalt_lower_outOfBounds_to_outOfBounds() (gas: 476148)
[FAIL] test_s2_originSwap_partialUpperAndLowerSlippage_balanced_40USDC_to_DAI()
```

```
[FAIL] test_s2_originSwap_noSlippage_balanced_30PctWeight_to_30PctWeight()
[FAIL] test_s2_originSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_CUSDC_to_
[OOPS] testFailOriginSwap_greaterThanBalance_10Pct()
[FAIL] test_s2_originSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight_to_10PctWe
[OOPS] testFailOriginSwap_greaterThanBalance_30Pct()
[FAIL] test_s2_originSwap_noSlippage_lightlyUnbalanced_10USDC_to_USDT_with_80DAI_100U
[PASS] test_s2_originSwap_smartHalt_lower_unrelated() (gas: 404117)
[PASS] test_s2_originSwap_upperHaltCheck_10PctWeight() (gas: 330070)
[PASS] test_s2_originSwap_smartHalt_upper() (gas: 384590)
[FAIL] test_s2_originSwap_partialUpperAndLowerSlippage_unbalanced_10PctWeight_to_30Pc
[FAIL] test_s2_originSwap_megaLowerToUpper_10PctWeight_to_30PctWeight()
[FAIL] test_s2_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight__()
[FAIL] test_s2_originSwap_megaUpperToLower_30PctWeight_to_10PctWeight()
[FAIL] test_s2_originSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight()
[FAIL] test_s2_originSwap_noSlippage_balanced_10PctWeight_to_30PctWeight()
[FAIL] test_s2_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_10PctWeight_to_
[FAIL] test_s2_originSwap_fullUpperAndLowerAntiSlippage_10PctWeight_to30PctWeight()
[PASS] test_s2_originSwap_smartHalt_lower_outOfBounds_to_inBounds() (gas: 468920)
[PASS] test_s2_originSwap_smartHalt_upper_unrelated() (gas: 404139)
[FAIL] test_s2_originSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_to_10PctW
[FAIL] test_s2_originSwap_megaLowerToUpperUpperToLower_30PctWeight()
[FAIL] test_s2_originSwap_megaLowerToUpperUpperToLower_CDAI_30PctWeight()
[PASS] test_s2_originSwap_lowerhaltCheck_10PctWeight() (gas: 340551)
[FAIL] test_s2_originSwap_fullUpperAndLowerSlippage_CUSDC_ASUSD_unbalanced_10PctWeigh
[PASS] test_s2_originSwap_upperHaltCheck_30PctWeight() (gas: 351009)
[FAIL] test_s2_originSwap_CHAI_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct()
[FAIL] test_s2_originSwap_fullUpperAndLowerSlippage_unbalanced_10PctWeight_to_30PctWe
[FAIL] test_s2_originSwap_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight()
[FAIL] test_s2_originSwap_fullUpperAndLowerAntiSlippage_unbalanced_30PctWeight()
[FAIL] test_s2_originSwap_noSlippage_balanced_10DAI_to_USDC_300Proportional()

Failure: test_s2_originSwap_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct

  "Error: Wrong `uint' value"
    Expected: 2365464484251272960
      Actual: 2365462191783708874


Failure: test_s2_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight_t

  "Error: Wrong `uint' value"
    Expected: 10019788191004510065
      Actual: 10019781368105515575


Failure: test_s2_originSwap_partialUpperAndLowerSlippage_balanced_40USDC_to_DAI

  "Error: Wrong `uint' value"
    Expected: 39330195827959985796
      Actual: 39329918950358908168
```

```
Failure: test_s2_originSwap_noSlippage_balanced_30PctWeight_to_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 9995000
      Actual: 9994999


Failure: test_s2_originSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_CUSDC_t

  "Error: Wrong `uint' value"
    Expected: 39330195827959985796
      Actual: 39329918023409094117


VM error for testFailOriginSwap_greaterThanBalance_10Pct()
Failure: test_s2_originSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight_to_10Pct

  "Error: Wrong `uint' value"
    Expected: 2876384124908864750
      Actual: 2876397434254408549


VM error for testFailOriginSwap_greaterThanBalance_30Pct()
Failure: test_s2_originSwap_noSlippage_lightlyUnbalanced_10USDC_to_USDT_with_80DAI_10

  "Error: Wrong `uint' value"
    Expected: 9995000
      Actual: 9994999


Failure: test_s2_originSwap_partialUpperAndLowerSlippage_unbalanced_10PctWeight_to_30

  "Error: Wrong `uint' value"
    Expected: 7920411672881948283
      Actual: 7920405190646252921


Failure: test_s2_originSwap_megaLowerToUpper_10PctWeight_to_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 19990003
      Actual: 19989999


Failure: test_s2_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight__

  "Error: Wrong `uint' value"
    Expected: 30070278169642344458
      Actual: 30070254970238095207


Failure: test_s2_originSwap_megaUpperToLower_30PctWeight_to_10PctWeight
```

```
  "Error: Wrong `uint' value"
    Expected: 19990016481618381864
      Actual: 19989999999999999971


Failure: test_s2_originSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 4666173
      Actual: 4666201


Failure: test_s2_originSwap_noSlippage_balanced_10PctWeight_to_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 3998000
      Actual: 3997999


Failure: test_s2_originSwap_partialUpperAndLowerAntiSlippage_unbalanced_10PctWeight_t

  "Error: Wrong `uint' value"
    Expected: 10006174300378984359
      Actual: 10006170882352941166


Failure: test_s2_originSwap_fullUpperAndLowerAntiSlippage_10PctWeight_to30PctWeight

  "Error: Wrong `uint' value"
    Expected: 3660153
      Actual: 3660152


Failure: test_s2_originSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_to_10Pc

  "Error: Wrong `uint' value"
    Expected: 14813513177462324025
      Actual: 14813392900989568306


Failure: test_s2_originSwap_megaLowerToUpperUpperToLower_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 69965119
      Actual: 69964999
  gas: 351469


Failure: test_s2_originSwap_megaLowerToUpperUpperToLower_CDAI_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 17401270
```

```
    Expected: 17491279
      Actual: 69964999


Failure: test_s2_originSwap_fullUpperAndLowerSlippage_CUSDC_ASUSD_unbalanced_10PctWei

  "Error: Wrong `uint' value"
    Expected: 2696349000000000000
      Actual: 2696359999999999999


Failure: test_s2_originSwap_CHAI_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct

  "Error: Wrong `uint' value"
    Expected: 2365464484251272960
      Actual: 2365462191783708874


Failure: test_s2_originSwap_fullUpperAndLowerSlippage_unbalanced_10PctWeight_to_30Pct

  "Error: Wrong `uint' value"
    Expected: 2696349
      Actual: 2696361


Failure: test_s2_originSwap_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight

  "Error: Wrong `uint' value"
    Expected: 2998500187500000000
      Actual: 2998499999999999997


Failure: test_s2_originSwap_fullUpperAndLowerAntiSlippage_unbalanced_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 5045804
      Actual: 5045796


Failure: test_s2_originSwap_noSlippage_balanced_10DAI_to_USDC_300Proportional

  "Error: Wrong `uint' value"
    Expected: 9995000
      Actual: 9994999


Running 34 tests for src/test/originSwaps/views/suiteOneViews.t.sol:OriginSwapiViewsS
[FAIL] test_s1_originSwapViews_fullUpperAndLowerSlippage_CUSDC_ASUSD_unbalanced_10Pct
[PASS] test_s1_originSwapViews_smartHalt_upper() (gas: 384593)
[PASS] test_s1_originSwapViews_noSlippage_balanced_10DAI_to_USDC_300Proportional() (g
[PASS] test_s1_originSwapViews_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeigh
[PASS] test_s1_originSwapViews_megaLowerToUpper_10PctWeight_to_30PctWeight() (gas: 31
[PASS] test_s1_originSwapViews_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct() (
```

```
[PASS] test_s1_originSwapViews_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pct() (
[PASS] test_s1_originSwapViews_smartHalt_lower_outOfBounds_to_inBounds() (gas: 468922
[PASS] test_s1_originSwapViews_smartHalt_lower_unrelated() (gas: 404141)
[PASS] test_s1_originSwapViews_noSlippage_lightlyUnbalanced_10USDC_to_USDT_with_80DAI
[PASS] test_s1_originSwapViews_fullUpperAndLowerAntiSlippage_unbalanced_30PctWeight()
[PASS] test_s1_originSwapViews_fullUpperAndLowerSlippage_unbalanced_30PctWeight_to_10
[FAIL] test_s1_originSwapViews_partialUpperAndLowerSlippage_balanced_30PctWeight_CUSD
[OOPS] testFailOriginSwap_greaterThanBalance_10Pct()
[PASS] test_s1_originSwapViews_noSlippage_balanced_30PctWeight_to_30PctWeight() (gas:
[PASS] test_s1_originSwapViews_megaLowerToUpperUpperToLower_30PctWeight() (gas: 31504
[OOPS] testFailOriginSwap_greaterThanBalance_30Pct()
[PASS] test_s1_originSwapViews_megaUpperToLower_30PctWeight_to_10PctWeight() (gas: 31
[PASS] test_s1_originSwapViews_lowerhaltCheck_10PctWeight() (gas: 340485)
[PASS] test_s1_originSwapViews_fullUpperAndLowerSlippage_unbalanced_30PctWeight__() (
[PASS] test_s1_originSwapViews_noSlippage_balanced_10PctWeight_to_30PctWeight() (gas:
[FAIL] test_s1_originSwapViews_partialUpperAndLowerSlippage_balanced_30PctWeight_to_1
[PASS] test_s1_originSwapViews_fullUpperAndLowerAntiSlippage_10PctWeight_to30PctWeigh
[PASS] test_s1_originSwapViews_upperHaltCheck_30PctWeight() (gas: 351054)
[PASS] test_s1_originSwapViews_partialUpperAndLowerSlippage_balanced_40USDC_to_DAI()
[FAIL] test_s1_originSwapViews_CHAI_fullUpperAndLowerAntiSlippage_30pctWeight_to_10Pc
[FAIL] test_s1_originSwapViews_megaLowerToUpperUpperToLower_CDAI_30PctWeight()
[PASS] test_s1_originSwapViews_smartHalt_lower_outOfBounds_to_outOfBounds() (gas: 476
[PASS] test_s1_originSwapViews_lowerHaltCheck_30PctWeight() (gas: 354222)
[PASS] test_s1_originSwapViews_smartHalt_upper_unrelated() (gas: 404183)
[PASS] test_s1_originSwapViews_fullUpperAndLowerSlippage_unbalanced_10PctWeight_to_30
[PASS] test_s1_originSwapViews_partialUpperAndLowerSlippage_unbalanced_10PctWeight_to
[PASS] test_s1_originSwapViews_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeigh
[PASS] test_s1_originSwapViews_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeigh
[PASS] test_s1_originSwapViews_upperHaltCheck_10PctWeight() (gas: 330067)

Failure: test_s1_originSwapViews_fullUpperAndLowerSlippage_CUSDC_ASUSD_unbalanced_10P

  "Error: Wrong `uint' value"
    Expected: 2696349000000000000
      Actual: 148121999999999999


Failure: test_s1_originSwapViews_partialUpperAndLowerSlippage_balanced_30PctWeight_CU

  "Error: Wrong `uint' value"
    Expected: 39339756348795716299
      Actual: 2167860899147103105


VM error for testFailOriginSwap_greaterThanBalance_10Pct()
VM error for testFailOriginSwap_greaterThanBalance_30Pct()
Failure: test_s1_originSwapViews_partialUpperAndLowerSlippage_balanced_30PctWeight_to

  "Error: Wrong `uint' value"
    Expected: 14817098063643470308
      Actual: 14817098101815228528
```

```
Failure: test_s1_originSwapViews_CHAI_fullUpperAndLowerAntiSlippage_30pctWeight_to_10

  "Error: Wrong `uint' value"
    Expected: 2366053853162344119
      Actual: 128540085137972614


Failure: test_s1_originSwapViews_megaLowerToUpperUpperToLower_CDAI_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 17491279
      Actual: 69982499


Running 8 tests for src/test/targetSwaps/suiteFive.t.sol:TargetSwapSuiteFiveTest
[OOPS] test_s5_targetSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuallyInBou
[OOPS] test_s5_targetSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuallyOutOf
[OOPS] test_s5_targetSwap_monotonicity_outOfBand_mutuallyOutOfBound_zero_noHalts_omeg
[OOPS] test_s5_targetSwap_monotonicity_mutuallyInBounds_to_mutuallyOutOfBounds_noHalt
[OOPS] test_s5_targetSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuallyOutOf
[PASS] test_s5_targetSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuallyInBou
[OOPS] test_s5_targetSwap_monotonicity_outOfBand_mutuallyOutOfBound_towards_mutuallyI
[OOPS] test_s5_targetSwap_monotonicity_mutuallyInBounds_to_mutuallyOutOfBounds_halts(

VM error for test_s5_targetSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuall
VM error for test_s5_targetSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuall
VM error for test_s5_targetSwap_monotonicity_outOfBand_mutuallyOutOfBound_zero_noHalt
VM error for test_s5_targetSwap_monotonicity_mutuallyInBounds_to_mutuallyOutOfBounds_
VM error for test_s5_targetSwap_monotonicity_outOfBand_mutuallyOutOfBounds_to_mutuall
VM error for test_s5_targetSwap_monotonicity_outOfBand_mutuallyOutOfBound_towards_mut
VM error for test_s5_targetSwap_monotonicity_mutuallyInBounds_to_mutuallyOutOfBounds_
Running 36 tests for src/test/targetSwaps/suiteOne.t.sol:TargetSwapSuiteOneTests
[PASS] test_s1_targetSwap_smartHalt_lower() (gas: 453173)
[PASS] test_s1_targetSwap_megaLowerToUpperUpperToLower_30PctWeight() (gas: 347050)
[PASS] test_s1_targetSwap_noSlippage_unbalanced_USDC_to_3SUSD_with_80DAI_100USDC_85US
[PASS] test_s1_targetSwap_noSlippage_Balanced_10PctWeight_to_30PctWeight_AUSDT() (gas
[PASS] test_s1_targetSwap_fullUpperAndLowerAntiSlippage_CDAI_30pct_to_10Pct() (gas: 4
[PASS] test_s1_targetSwap_megaUpperToLower_30PctWeight_to_10PctWeight() (gas: 346272)
[PASS] test_s1_targetSwap_lowerHaltCheck_30PctWeight() (gas: 343296)
[PASS] test_s1_targetSwap_lowerhaltCheck_10PctWeight() (gas: 323458)
[PASS] test_s1_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_10PctWeight_to_
[OOPS] testFailTargetSwap_targetGreaterThanBalance_30Pct()
[PASS] test_s1_targetSwap_smartHalt_lower_unrelated() (gas: 453241)
[PASS] test_s1_targetSwap_fullUpperAndLowerAntiSlippage_10PctOrigin_to_30PctTarget()
[PASS] test_s1_targetSwap_noSlippage_balanced_DAI_to_10USDC_300Proportional() (gas: 3
[PASS] test_s1_targetSwap_noSlippage_balanced_10PctWeight_to_30PctWeight() (gas: 3336
[PASS] test_s1_targetSwap_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight() (
[PASS] test_s1_targetSwap_smartHalt_upper_outOfBounds_to_inBounds() (gas: 393025)
[OOPS] testFailTargetSwap_targetGreaterThanBalance_10Pct()
[PASS] test_s1_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_CHAI_10PctWeigh
[PASS] test_s1_targetSwap_upperHaltCheck_10PctWeight() (gas: 278306)
[PASS] test_s1_targetSwap_fullUpperAndLowerSlippage_unbalanced_10PctWeight_to_30PctWe
```

```
[PASS] test_s1_targetSwap_smartHalt_upper_unrelated() (gas: 404933)
[PASS] test_s1_targetSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_to_10PctW
[PASS] test_s1_targetSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_to30PctWe
[PASS] test_s1_targetSwap_megaLowerToUpper_10PctWeight_to_30PctWeight() (gas: 346138)
[PASS] test_s1_targetSwap_fullUpperAndLowerAntiSlippage_30Pct_To10Pct() (gas: 355420)
[PASS] test_s1_targetSwap_upperHaltCheck_30PctWeight() (gas: 343742)
[PASS] test_s1_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight_to_
[PASS] test_s1_targetSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight_to_10PctWe
[PASS] test_s1_targetSwap_noSlippage_partiallyUnbalanced_10PctTarget() (gas: 345511)
[PASS] test_s1_targetSwap_partialUpperAndLowerSLippage_balanced_30PctWeight_to_10PctW
[PASS] test_s1_targetSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight() (gas: 36
[PASS] test_s1_targetSwap_fullUpperAndLowerAntiSlippage_unbalanced_30PctWeight() (gas
[PASS] test_s1_targetSwap_smartHalt_upper_outOfBounds_to_outOfBounds() (gas: 397936)
[PASS] test_s1_targetSwap_partialUpperAndLowerSlippage_unbalanced_10PctWeight_to_30Pc
[PASS] test_s1_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight_to_
[PASS] test_s1_targetSwap_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight_CUS

VM error for testFailTargetSwap_targetGreaterThanBalance_30Pct()
VM error for testFailTargetSwap_targetGreaterThanBalance_10Pct()
Running 3 tests for src/test/targetSwaps/suiteSix.t.sol:TargetSwapSuiteSixTest
[PASS] test_s6_targetSwap_continuity_antiSlippage() (gas: 801239)
[PASS] test_s6_targetSwap_continuity_slippage() (gas: 802912)
[PASS] test_s6_targetSwap_continuity_balanced() (gas: 762100)

Running 31 tests for src/test/targetSwaps/suiteTwo.t.sol:TargetSwapSuiteTwoTests
[PASS] test_s2_targetSwap_upperHaltCheck_10PctWeight() (gas: 278308)
[FAIL] test_s2_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_10PctWeight_to_
[FAIL] test_s2_targetSwap_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight_CUS
[PASS] test_s2_targetSwap_noSlippage_unbalanced_USDC_to_3SUSD_with_80DAI_100USDC_85US
[PASS] test_s2_targetSwap_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight() (
[FAIL] test_s2_targetSwap_noSlippage_partiallyUnbalanced_10PctTarget()
[FAIL] test_s2_targetSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight_to_10PctWe
[OOPS] testFailTargetSwap_targetGreaterThanBalance_30Pct()
[FAIL] test_s2_targetSwap_megaLowerToUpper_10PctWeight_to_30PctWeight()
[PASS] test_s2_targetSwap_upperHaltCheck_30PctWeight() (gas: 343742)
[FAIL] test_s2_targetSwap_partialUpperAndLowerSlippage_unbalanced_10PctWeight_to_30Pc
[PASS] test_s2_targetSwap_lowerhaltCheck_10PctWeight() (gas: 323414)
[PASS] test_s2_targetSwap_lowerHaltCheck_30PctWeight() (gas: 343273)
[FAIL] test_s2_targetSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight()
[FAIL] test_s2_targetSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_to_10PctW
[OOPS] testFailTargetSwap_targetGreaterThanBalance_10Pct()
[FAIL] test_s2_targetSwap_megaLowerToUpperUpperToLower_30PctWeight()
[FAIL] test_s2_targetSwap_noSlippage_balanced_DAI_to_10USDC_300Proportional()
[FAIL] test_s2_targetSwap_fullUpperAndLowerAntiSlippage_30Pct_To10Pct()
[FAIL] test_s2_targetSwap_fullUpperAndLowerAntiSlippage_unbalanced_30PctWeight()
[FAIL] test_s2_targetSwap_partialUpperAndLowerSLippage_balanced_30PctWeight_to_10PctW
[FAIL] test_s2_targetSwap_fullUpperAndLowerSlippage_unbalanced_10PctWeight_to_30PctWe
[FAIL] test_s2_targetSwap_fullUpperAndLowerAntiSlippage_CDAI_30pct_to_10Pct()
[FAIL] test_s2_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_CHAI_10PctWeigh
[FAIL] test_s2_targetSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_to30PctWe
[FAIL] test_s2_targetSwap_fullUpperAndLowerAntiSlippage_10PctOrigin_to_30PctTarget()
[FAIL] test_s2_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight_to_
```

```
[FAIL] test_s2_targetSwap_noSlippage_Balanced_10PctWeight_to_30PctWeight_AUSDT()
[FAIL] test_s2_targetSwap_megaUpperToLower_30PctWeight_to_10PctWeight()
[FAIL] test_s2_targetSwap_noSlippage_balanced_10PctWeight_to_30PctWeight()
[FAIL] test_s2_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight_to_

Failure: test_s2_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_10PctWeight_t

  "Error: Wrong `uint' value"
    Expected: 9993821361386267461
      Actual: 9993817941176470000


Failure: test_s2_targetSwap_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight_C

  "Error: Wrong `uint' value"
    Expected: 3001500000000000000
      Actual: 3001498999999999999


Failure: test_s2_targetSwap_noSlippage_partiallyUnbalanced_10PctTarget

  "Error: Wrong `uint' value"
    Expected: 3001500187500000000
      Actual: 3001500000000000000


Failure: test_s2_targetSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight_to_10Pct

  "Error: Wrong `uint' value"
    Expected: 3130264791663764854
      Actual: 3130274781523582000


VM error for testFailTargetSwap_targetGreaterThanBalance_30Pct()
Failure: test_s2_targetSwap_megaLowerToUpper_10PctWeight_to_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 20010074968656541264
      Actual: 20010000000000000000


Failure: test_s2_targetSwap_partialUpperAndLowerSlippage_unbalanced_10PctWeight_to_30

  "Error: Wrong `uint' value"
    Expected: 8082681715960427072
      Actual: 8082647704924231000


Failure: test_s2_targetSwap_fullUpperAndLowerSlippage_unbalanced_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 5361455914007417759
```

```
      Actual: 5292593994805449000


Failure: test_s2_targetSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_to_10Pc

  "Error: Wrong `uint' value"
    Expected: 12073660
      Actual: 12073670


VM error for testFailTargetSwap_targetGreaterThanBalance_10Pct()
Failure: test_s2_targetSwap_megaLowerToUpperUpperToLower_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 70035406577130885767
      Actual: 70035000000000000000


Failure: test_s2_targetSwap_noSlippage_balanced_DAI_to_10USDC_300Proportional

  "Error: Wrong `uint' value"
    Expected: 10005000625000000000
      Actual: 10005000000000000000


Failure: test_s2_targetSwap_fullUpperAndLowerAntiSlippage_30Pct_To10Pct

  "Error: Wrong `uint' value"
    Expected: 2332615973232859927
      Actual: 2332612242748136000


Failure: test_s2_targetSwap_fullUpperAndLowerAntiSlippage_unbalanced_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 4954524
      Actual: 4954516


Failure: test_s2_targetSwap_partialUpperAndLowerSLippage_balanced_30PctWeight_to_10Pc

  "Error: Wrong `uint' value"
    Expected: 12073660
      Actual: 12073670


Failure: test_s2_targetSwap_fullUpperAndLowerSlippage_unbalanced_10PctWeight_to_30Pct

  "Error: Wrong `uint' value"
    Expected: 2909155536050677534
      Actual: 2909159861885159000
```

```
Failure: test_s2_targetSwap_fullUpperAndLowerAntiSlippage_CDAI_30pct_to_10Pct

  "Error: Wrong `uint' value"
    Expected: 2332615973198180868
      Actual: 2332612242729300572


Failure: test_s2_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_CHAI_10PctWei

  "Error: Wrong `uint' value"
    Expected: 9993821361386267461
      Actual: 9993817941176470000


Failure: test_s2_targetSwap_partialUpperAndLowerSlippage_balanced_30PctWeight_to30Pct

  "Error: Wrong `uint' value"
    Expected: 40722871
      Actual: 40722721


Failure: test_s2_targetSwap_fullUpperAndLowerAntiSlippage_10PctOrigin_to_30PctTarget

  "Error: Wrong `uint' value"
    Expected: 3647253554589698680
      Actual: 3647251783776860000


Failure: test_s2_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight_t

  "Error: Wrong `uint' value"
    Expected: 29929682
      Actual: 29929659


Failure: test_s2_targetSwap_noSlippage_Balanced_10PctWeight_to_30PctWeight_AUSDT

  "Error: Wrong `uint' value"
    Expected: 4002000250000000000
      Actual: 4002000000000000000


Failure: test_s2_targetSwap_megaUpperToLower_30PctWeight_to_10PctWeight

  "Error: Wrong `uint' value"
    Expected: 20010007164941759473
      Actual: 20010000000000000000


Failure: test_s2_targetSwap_noSlippage_balanced_10PctWeight_to_30PctWeight
```

```
  "Error: Wrong `uint' value"
    Expected: 4002000250000000000
      Actual: 4002000000000000000


Failure: test_s2_targetSwap_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight_t

  "Error: Wrong `uint' value"
    Expected: 9980200
      Actual: 9980193


Running 36 tests for src/test/targetSwaps/views/suiteOneViews.t.sol:TargetSwapViewsSu
[PASS] test_s1_targetSwapView_partialUpperAndLowerSlippage_balanced_30PctWeight_to_10
[FAIL] test_s1_targetSwapView_fullUpperAndLowerSlippage_unbalanced_30PctWeight_to_10P
[FAIL] test_s1_targetSwapView_noSlippage_balanced_DAI_to_10USDC_300Proportional()
[FAIL] test_s1_targetSwapView_fullUpperAndLowerSlippage_unbalanced_10PctWeight_to_30P
[PASS] test_s1_targetSwapView_smartHalt_upper_outOfBounds_to_outOfBounds() (gas: 3979
[PASS] test_s1_targetSwapView_smartHalt_upper_outOfBounds_to_inBounds() (gas: 393049)
[FAIL] test_s1_targetSwapView_partialUpperAndLowerAntiSlippage_unbalanced_10PctWeight
[OOPS] testFailTargetSwap_targetGreaterThanBalance_30Pct()
[FAIL] test_s1_targetSwapView_fullUpperAndLowerAntiSlippage_10PctOrigin_to_30PctTarge
[FAIL] test_s1_targetSwapView_megaLowerToUpper_10PctWeight_to_30PctWeight()
[PASS] test_s1_targetSwapView_smartHalt_upper_unrelated() (gas: 404889)
[FAIL] test_s1_targetSwapView_noSlippage_balanced_10PctWeight_to_30PctWeight()
[PASS] test_s1_targetSwapView_upperHaltCheck_10PctWeight() (gas: 278306)
[PASS] test_s1_targetSwapView_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight
[PASS] test_s1_targetSwapView_fullUpperAndLowerAntiSlippage_CDAI_30pct_to_10Pct() (ga
[PASS] test_s1_targetSwapView_partialUpperAndLowerSLippage_balanced_30PctWeight_to_10
[PASS] test_s1_targetSwapView_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight
[PASS] test_s1_targetSwapView_smartHalt_lower_unrelated() (gas: 453219)
[PASS] test_s1_targetSwapView_noSlippage_unbalanced_USDC_to_3SUSD_with_80DAI_100USDC_
[OOPS] testFailTargetSwap_targetGreaterThanBalance_10Pct()
[PASS] test_s1_targetSwapView_upperHaltCheck_30PctWeight() (gas: 343721)
[PASS] test_s1_targetSwapView_partialUpperAndLowerAntiSlippage_unbalanced_30PctWeight
[FAIL] test_s1_targetSwapView_megaUpperToLower_30PctWeight_to_10PctWeight()
[PASS] test_s1_targetSwapView_fullUpperAndLowerAntiSlippage_unbalanced_30PctWeight()
[PASS] test_s1_targetSwapView_smartHalt_lower() (gas: 453236)
[FAIL] test_s1_targetSwapView_noSlippage_partiallyUnbalanced_10PctTarget()
[FAIL] test_s1_targetSwapView_megaLowerToUpperUpperToLower_30PctWeight()
[FAIL] test_s1_targetSwapView_partialUpperAndLowerSlippage_unbalanced_10PctWeight_to_
[OOPS] test_s1_targetSwapView_noSlippage_Balanced_10PctWeight_to_30PctWeight_AUSDT()
[FAIL] test_s1_targetSwapView_partialUpperAndLowerAntiSlippage_unbalanced_CHAI_10PctW
[FAIL] test_s1_targetSwapView_fullUpperAndLowerAntiSlippage_30Pct_To10Pct()
[PASS] test_s1_targetSwapView_partialUpperAndLowerSlippage_balanced_30PctWeight_to30P
[FAIL] test_s1_targetSwapView_fullUpperAndLowerSlippage_unbalanced_30PctWeight()
[PASS] test_s1_targetSwapView_lowerhaltCheck_10PctWeight() (gas: 323434)
[FAIL] test_s1_targetSwapView_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeight
[PASS] test_s1_targetSwapView_lowerHaltCheck_30PctWeight() (gas: 343293)

Failure: test_s1_targetSwapView_fullUpperAndLowerSlippage_unbalanced_30PctWeight_to_1
```

```
  "Error: Wrong `uint' value"
    Expected: 3129492601572409000
      Actual: 3129492603917005095


Failure: test_s1_targetSwapView_noSlippage_balanced_DAI_to_10USDC_300Proportional

  "Error: Wrong `uint' value"
    Expected: 10002500000000000000
      Actual: 10002500000000000009


Failure: test_s1_targetSwapView_fullUpperAndLowerSlippage_unbalanced_10PctWeight_to_3

  "Error: Wrong `uint' value"
    Expected: 2908432935382939000
      Actual: 2908432935382939065


Failure: test_s1_targetSwapView_partialUpperAndLowerAntiSlippage_unbalanced_10PctWeig

  "Error: Wrong `uint' value"
    Expected: 9991320735294117000
      Actual: 9991320735294117657


VM error for testFailTargetSwap_targetGreaterThanBalance_30Pct()
Failure: test_s1_targetSwapView_fullUpperAndLowerAntiSlippage_10PctOrigin_to_30PctTar

  "Error: Wrong `uint' value"
    Expected: 3646340429241883000
      Actual: 3646340426509550281


Failure: test_s1_targetSwapView_megaLowerToUpper_10PctWeight_to_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 20005000000000000000
      Actual: 20005000000000000016


Failure: test_s1_targetSwapView_noSlippage_balanced_10PctWeight_to_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 4001000000000000000
      Actual: 4001000000000000003


VM error for testFailTargetSwap_targetGreaterThanBalance_10Pct()
Failure: test_s1_targetSwapView_megaUpperToLower_30PctWeight_to_10PctWeight

  "Error: Wrong `uint' value"
```

```
    Expected: 20005000000000000000
      Actual: 20005000000000000028


Failure: test_s1_targetSwapView_noSlippage_partiallyUnbalanced_10PctTarget

  "Error: Wrong `uint' value"
    Expected: 3000750000000000000
      Actual: 3000750000000000002


Failure: test_s1_targetSwapView_megaLowerToUpperUpperToLower_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 70017500000000000000
      Actual: 70017500000000000064


Failure: test_s1_targetSwapView_partialUpperAndLowerSlippage_unbalanced_10PctWeight_t

  "Error: Wrong `uint' value"
    Expected: 8080628052824050000
      Actual: 8080628052824050355


VM error for test_s1_targetSwapView_noSlippage_Balanced_10PctWeight_to_30PctWeight_AU
Failure: test_s1_targetSwapView_partialUpperAndLowerAntiSlippage_unbalanced_CHAI_10Pc

  "Error: Wrong `uint' value"
    Expected: 9991320735294117000
      Actual: 9991320735294117657


Failure: test_s1_targetSwapView_fullUpperAndLowerAntiSlippage_30Pct_To10Pct

  "Error: Wrong `uint' value"
    Expected: 2332029381118264000
      Actual: 2332029381118264742


Failure: test_s1_targetSwapView_fullUpperAndLowerSlippage_unbalanced_30PctWeight

  "Error: Wrong `uint' value"
    Expected: 5291271507550375000
      Actual: 5291271507550375805


Failure: test_s1_targetSwapView_noSlippage_lightlyUnbalanced_30PctWeight_to_10PctWeig

  "Error: Wrong `uint' value"
    Expected: 3001500000000000000
      Actual: 3000748999999999999
```

```
Running 19 tests for src/test/testAssimilators.t.sol:AssimilatorSetOneTests
[PASS] testAssimilator_USDC_to_CUSDC_views() (gas: 7329)
[OOPS] testAssimilator_CHAI_to_CDAI_raws()
[PASS] testAssimilator_CDAI_to_CDAI_views() (gas: 12966)
[PASS] testAssimilator_SUSD_to_ASUSD() (gas: 279)
[OOPS] testAssimilator_CUSDC_to_CUSDC_numeraires()
[PASS] testAssimilator_CUSDC_to_CUSDC_views() (gas: 12955)
[PASS] testAssimilator_CHAI_to_CDAI_views() (gas: 13461)
[PASS] testAssimilator_ASUSD_to_ASUSD() (gas: 256)
[PASS] testAssimilator_USDT_to_AUSDT() (gas: 233)
[PASS] testAssimilator_CHAI_to_CDAI_numeraires() (gas: 77915)
[OOPS] testAssimilator_DAI_to_CDAI_raws()
[PASS] testAssimilator_USDC_to_CUSDC_numeraires() (gas: 212)
[OOPS] testAssimilator_CUSDC_to_CUSDC_raws()
[OOPS] testAssimilator_CDAI_to_CDAI_numeraires()
[PASS] testAssimilator_AUSDT_to_AUSDT() (gas: 234)
[PASS] testAssimilator_DAI_to_CDAI_numeraires() (gas: 54385)
[OOPS] testAssimilator_CDAI_to_CDAI_raws()
[OOPS] testAssimilator_USDC_to_CUSDC_raws()
[PASS] testAssimilator_DAI_to_CDAI_views() (gas: 7337)

VM error for testAssimilator_CHAI_to_CDAI_raws()
VM error for testAssimilator_CUSDC_to_CUSDC_numeraires()
VM error for testAssimilator_DAI_to_CDAI_raws()
VM error for testAssimilator_CUSDC_to_CUSDC_raws()
VM error for testAssimilator_CDAI_to_CDAI_numeraires()
VM error for testAssimilator_CDAI_to_CDAI_raws()
VM error for testAssimilator_USDC_to_CUSDC_raws()
Running 9 tests for src/test/testAssimilators.t.sol:AssimilatorSetTwoTests
[PASS] testAssimilator_AUSDT_to_USDT() (gas: 279)
[PASS] testAssimilator_DAI_to_DAI() (gas: 256)
[PASS] testAssimilator_SUSD_to_SUSD() (gas: 234)
[PASS] testAssimilator_USDT_to_USDT() (gas: 233)
[PASS] testAssimilator_CHAI_to_DAI() (gas: 190)
[PASS] testAssimilator_CUSDC_to_USDC() (gas: 278)
[PASS] testAssimilator_ASUSD_to_SUSD() (gas: 300)
[PASS] testAssimilator_CDAI_to_DAI() (gas: 256)
[PASS] testAssimilator_USDC_to_USDC() (gas: 255)

Running 16 tests for src/test/withdraws/suiteFive.t.sol:SelectiveWithdrawSuiteFive
[PASS] test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_outOfBou
[OOPS] test_s5_selectiveWithdraw_monotonicity_lower_outOfBand_outOfBounds_to_outOfBou
[OOPS] test_s5_selectiveWithdraw_monotonicity_lower_inBounds_to_outOfBounds_noHalt()
[OOPS] test_s5_proportionalWithdraw_monotonicity_upper_outOfBand()
[OOPS] test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_inBounds
[OOPS] test_s5_selectiveWithdraw_monotonicity_lower_inBounds_to_outOfBounds_halt()
[PASS] test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_inBounds
[OOPS] test_s5_selectiveWithdraw_monotonicity_lower_outOfBand_outOfBounds_to_outOfBou
[PASS] test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_inBounds
[OOPS] test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_inBounds
[OOPS] test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_outOfBou
```

```
[OOPS] test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_outOfBou
[OOPS] test_s5_selectiveWithdraw_monotonicity_lower_outOfBand_outOfBounds_to_outOfBou
[OOPS] test_s5_proportionalWithdraw_monotonicity_lower_outOfBand()
[OOPS] test_s5_selectiveWithdraw_monotonicity_lower_outOfBand_outOfBounds_to_outOfBou
[PASS] test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_outOfBou
[OOPS] test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_outOfBou

VM error for test_s5_selectiveWithdraw_monotonicity_lower_outOfBand_outOfBounds_to_ou
VM error for test_s5_selectiveWithdraw_monotonicity_lower_inBounds_to_outOfBounds_noH
VM error for test_s5_proportionalWithdraw_monotonicity_upper_outOfBand()
VM error for test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_in
VM error for test_s5_selectiveWithdraw_monotonicity_lower_inBounds_to_outOfBounds_hal
VM error for test_s5_selectiveWithdraw_monotonicity_lower_outOfBand_outOfBounds_to_ou
VM error for test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_in
VM error for test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_ou
VM error for test_s5_selectiveWithdraw_monotonicity_lower_outOfBand_outOfBounds_to_ou
VM error for test_s5_proportionalWithdraw_monotonicity_lower_outOfBand()
VM error for test_s5_selectiveWithdraw_monotonicity_lower_outOfBand_outOfBounds_to_ou
VM error for test_s5_selectiveWithdraw_monotonicity_upper_outOfBand_outOfBounds_to_ou
Running 30 tests for src/test/withdraws/suiteOne.t.sol:SelectiveWithdrawSuiteOne
[PASS] test_s1_selectiveWithdraw_smartHalt_upper_outOfBounds_to_inBounds() (gas: 3718
[PASS] test_s1_selectiveWithdraw_partialLowerAntiSlippage_0p0001DAI_41USDC_41USDT_1SU
[PASS] test_s1_selectiveWithdraw_partialLowerIndirectAntiSlippage_40DAI_40USDT_from_9
[PASS] test_s1_selectiveWithdraw_smartHalt_lower_outOfBounds_exacerbated() (gas: 4233
[OOPS] testFailSelectiveWithdraw_lowerHaltCheck30Pct()
[PASS] test_s1_selectiveWithdraw_fullLowerSlippage_1USDC_7USDT_2SUSD_from_95DAI_95USD
[PASS] test_s1_selectiveWithdraw_balanced_10DAI_10USDC_10USDT_2p5SUSD_from_300Proport
[PASS] test_s1_selectiveWithdraw_fullIndirectUpperSlippage_5DAI_5USDT_from90DAI_145US
[PASS] test_s1_selectiveWithdraw_smartHalt_upper_unrelated() (gas: 346378)
[PASS] test_s1_selectiveWithdraw_megaUpperToLower_95USDT_35SUSD_from_90DAI_90USDC_145
[PASS] test_s1_selectiveWithdraw_fullLowerAntiSlippageWithdraw_5DAI_5USDC_0p5USDT_0p2
[PASS] test_s1_selectiveWithdraw_lightlyUnbalanced_5DAI_1USDC_3USDT_1SUSD_from_80DAI_
[PASS] test_s1_selectiveWithdraw_fullUpperAntiSlippage_5DAI_2SUSD_from_145DAI_90USDC_
[PASS] test_s1_selectiveWithdraw_smartHalt_lower_outOfBounds_to_inBounds() (gas: 4736
[PASS] test_s1_selectiveWithdraw_partialLowerSlippage_3DAI_60USDC_30USDT_1SUSD_from_8
[PASS] test_s1_selectiveWithdraw_megaIndirectLowerToUpper_11DAI_74USDC_74USDT_from_55
[PASS] test_s1_selectiveWithdraw_megaIndirectWithdrawLowerToUpper_11DAI_74USDC_74USDT
[PASS] test_s1_selectiveWithdraw_partialLowerSlippage_balanced_5DAI_5USDC_47USDT_16SU
[OOPS] testFailSelectiveWithdraw_upperHaltCheck10Pct()
[PASS] test_s1_selectiveWithdraw_partialUpperAntiSlippage_50USDC_18SUSD_from_90DAI_14
[OOPS] test_s1_selectiveWithdraw_fullIndirectLowerAntiSlippage_5CHAI_5CUSDC_from_95DA
[OOPS] test_s1_selectiveWithdraw_fullUpperAntiSlippage_5CDAI_2ASUSD_from_145DAI_90USD
[PASS] test_s1_selectiveWithdraw_partialUpperSlippage_balanced_0p001DAI_40USDC_40USDT
[OOPS] testFailSelectiveWithdraw_lowerHaltCheck10Pct()
[PASS] test_s1_selectiveWithdraw_smartHalt_upper_outOfBounds_to_outOfBounds() (gas: 3
[PASS] test_s1_selectiveWithdraw_fullIndirectLowerAntiSlippage_5DAI_5USDC_from_95DAI_
[OOPS] test_s1_selectiveWithdraw_partialUpperAntiSlippage_50CUSDC_18SUSD_from_90DAI_1
[PASS] test_s1_selectiveWithdraw_smartHalt_lower_outOfBounds_to_outOfBounds() (gas: 4
[OOPS] testFailSelectiveWithdraw_upperHaltCheck30Pct()
[PASS] test_s1_selectiveWithdraw_fullUpperSlippage_8DAI_2USDC_8USDT_2SUSD_from_90DAI_

VM error for testFailSelectiveWithdraw_lowerHaltCheck30Pct()
VM error for testFailSelectiveWithdraw_upperHaltCheck10Pct()
```

```
VM error for testFailSelectiveWithdraw_upperHaltCheck10Pct()
VM error for test_s1_selectiveWithdraw_fullIndirectLowerAntiSlippage_5CHAI_5CUSDC_fro
VM error for test_s1_selectiveWithdraw_fullUpperAntiSlippage_5CDAI_2ASUSD_from_145DAI
VM error for testFailSelectiveWithdraw_lowerHaltCheck10Pct()
VM error for test_s1_selectiveWithdraw_partialUpperAntiSlippage_50CUSDC_18SUSD_from_9
VM error for testFailSelectiveWithdraw_upperHaltCheck30Pct()
Running 3 tests for src/test/withdraws/suiteSix.t.sol:SelectiveWithdrawSuiteSix
[PASS] test_s6_selectiveWithdraw_continuity_antiSlippage() (gas: 799220)
[PASS] test_s6_selectiveWithdraw_continuity_noSlippage_noAntiSlippage() (gas: 843955)
[PASS] test_s6_selectiveWithdraw_continuity_slippage() (gas: 845774)

Running 28 tests for src/test/withdraws/suiteTwo.t.sol:SelectiveWithdrawSuiteOne
[PASS] test_s2_selectiveWithdraw_smartHalt_outOfBounds_to_inBounds() (gas: 371916)
[OOPS] test_s2_selectiveWithdraw_fullIndirectLowerAntiSlippage_5CHAI_5CUSDC_from_95DA
[FAIL] test_s2_selectiveWithdraw_fullUpperAntiSlippage_5DAI_2SUSD_from_145DAI_90USDC_
[OOPS] testFailSelectiveWithdraw_lowerHaltCheck30Pct()
[OOPS] test_s2_selectiveWithdraw_partialUpperAntiSlippage_50CUSDC_18SUSD_from_90DAI_1
[FAIL] test_s2_selectiveWithdraw_partialUpperSlippage_balanced_0p001DAI_40USDC_40USDT
[FAIL] test_s2_selectiveWithdraw_fullLowerSlippage_1USDC_7USDT_2SUSD_from_95DAI_95USD
[PASS] test_s2_selectiveWithdraw_smartHalt_lower_outOfBounds_to_inBounds() (gas: 4735
[FAIL] test_s2_selectiveWithdraw_partialLowerAntiSlippage_0p0001DAI_41USDC_41USDT_1SU
[FAIL] test_s2_selectiveWithdraw_partialLowerSlippage_3DAI_60USDC_30USDT_1SUSD_from_8
[FAIL] test_s2_selectiveWithdraw_fullLowerAntiSlippageWithdraw_5DAI_5USDC_0p5USDT_0p2
[FAIL] test_s2_selectiveWithdraw_balanced_10DAI_10USDC_10USDT_2p5SUSD_from_300Proport
[FAIL] test_s2_selectiveWithdraw_fullIndirectUpperSlippage_5DAI_5USDT_from90DAI_145US
[FAIL] test_s2_selectiveWithdraw_fullUpperSlippage_8DAI_2USDC_8USDT_2SUSD_from_90DAI_
[FAIL] test_s2_selectiveWithdraw_lightlyUnbalanced_5DAI_1USDC_3USDT_1SUSD_from_80DAI_
[FAIL] test_s2_selectiveWithdraw_megaIndirectWithdrawLowerToUpper_11DAI_74USDC_74USDT
[FAIL] test_s2_selectiveWithdraw_partialUpperAntiSlippage_50USDC_18SUSD_from_90DAI_14
[OOPS] testFailSelectiveWithdraw_upperHaltCheck10Pct()
[OOPS] test_s2_selectiveWithdraw_fullUpperAntiSlippage_5CDAI_2ASUSD_from_145DAI_90USD
[OOPS] testFailSelectiveWithdraw_lowerHaltCheck10Pct()
[FAIL] test_s2_selectiveWithdraw_partialLowerIndirectAntiSlippage_40DAI_40USDT_from_9
[FAIL] test_s2_selectiveWithdraw_fullIndirectLowerAntiSlippage_5DAI_5USDC_from_95DAI_
[FAIL] test_s2_selectiveWithdraw_partialLowerSlippage_balanced_5DAI_5USDC_47USDT_16SU
[PASS] test_s2_selectiveWithdraw_smartHalt_lower_outOfBounds_to_outOfBounds() (gas: 4
[OOPS] testFailSelectiveWithdraw_upperHaltCheck30Pct()
[PASS] test_s2_selectiveWithdraw_smartHalt_outOfBounds_to_outOfBounds() (gas: 374273)
[FAIL] test_s2_selectiveWithdraw_megaUpperToLower_95USDT_35SUSD_from_90DAI_90USDC_145
[FAIL] test_s2_selectiveWithdraw_megaIndirectLowerToUpper_11DAI_74USDC_74USDT_from_55

VM error for test_s2_selectiveWithdraw_fullIndirectLowerAntiSlippage_5CHAI_5CUSDC_fro
Failure: test_s2_selectiveWithdraw_fullUpperAntiSlippage_5DAI_2SUSD_from_145DAI_90USD

  "Error: Wrong `uint' value"
    Expected: 6996035991529215020
      Actual: 6996036011473429940


VM error for testFailSelectiveWithdraw_lowerHaltCheck30Pct()
VM error for test_s2_selectiveWithdraw_partialUpperAntiSlippage_50CUSDC_18SUSD_from_9
Failure: test_s2_selectiveWithdraw_partialUpperSlippage_balanced_0p001DAI_40USDC_40US
```

```
  "Error: Wrong `uint' value"
    Expected: 90224422906045360592
      Actual: 90224421960738460186


Failure: test_s2_selectiveWithdraw_fullLowerSlippage_1USDC_7USDT_2SUSD_from_95DAI_95U

  "Error: Wrong `uint' value"
    Expected: 10134109814565570448
      Actual: 10134109817307692313


Failure: test_s2_selectiveWithdraw_partialLowerAntiSlippage_0p0001DAI_41USDC_41USDT_1

  "Error: Wrong `uint' value"
    Expected: 83002127076568926436
      Actual: 83002127396794871860


Failure: test_s2_selectiveWithdraw_partialLowerSlippage_3DAI_60USDC_30USDT_1SUSD_from

  "Error: Wrong `uint' value"
    Expected: 94102228495008790366
      Actual: 94102228808064194663


Failure: test_s2_selectiveWithdraw_fullLowerAntiSlippageWithdraw_5DAI_5USDC_0p5USDT_0

  "Error: Wrong `uint' value"
    Expected: 10696820295674489134
      Actual: 10696820295820476818


Failure: test_s2_selectiveWithdraw_balanced_10DAI_10USDC_10USDT_2p5SUSD_from_300Propo

  "Error: Wrong `uint' value"
    Expected: 32508125216729574694
      Actual: 32508125000000000018


Failure: test_s2_selectiveWithdraw_fullIndirectUpperSlippage_5DAI_5USDT_from90DAI_145

  "Error: Wrong `uint' value"
    Expected: 10072190169539376480
      Actual: 10072190173135464226


Failure: test_s2_selectiveWithdraw_fullUpperSlippage_8DAI_2USDC_8USDT_2SUSD_from_90DA

  "Error: Wrong `uint' value"
    Expected: 20090545586275051778
      Actual: 20090545637715179967
```

```
Failure: test_s2_selectiveWithdraw_lightlyUnbalanced_5DAI_1USDC_3USDT_1SUSD_from_80DA

  "Error: Wrong `uint' value"
    Expected: 10002499999733097916
      Actual: 10002500000000000000


Failure: test_s2_selectiveWithdraw_megaIndirectWithdrawLowerToUpper_11DAI_74USDC_74US

  "Error: Wrong `uint' value"
    Expected: 159145586630360938967
      Actual: 159145586600251986918


Failure: test_s2_selectiveWithdraw_partialUpperAntiSlippage_50USDC_18SUSD_from_90DAI_

  "Error: Wrong `uint' value"
    Expected: 68008386735015754177
      Actual: 68008386736111111158


VM error for testFailSelectiveWithdraw_upperHaltCheck10Pct()
VM error for test_s2_selectiveWithdraw_fullUpperAntiSlippage_5CDAI_2ASUSD_from_145DAI
VM error for testFailSelectiveWithdraw_lowerHaltCheck10Pct()
Failure: test_s2_selectiveWithdraw_partialLowerIndirectAntiSlippage_40DAI_40USDT_from

  "Error: Wrong `uint' value"
    Expected: 80001277060135043666
      Actual: 80001277371794871867


Failure: test_s2_selectiveWithdraw_fullIndirectLowerAntiSlippage_5DAI_5USDC_from_95DA

  "Error: Wrong `uint' value"
    Expected: 9995446955063918311
      Actual: 9995446955128205126


Failure: test_s2_selectiveWithdraw_partialLowerSlippage_balanced_5DAI_5USDC_47USDT_16

  "Error: Wrong `uint' value"
    Expected: 73154345690075849040
      Actual: 73154344955029368640


VM error for testFailSelectiveWithdraw_upperHaltCheck30Pct()
Failure: test_s2_selectiveWithdraw_megaUpperToLower_95USDT_35SUSD_from_90DAI_90USDC_1

  "Error: Wrong `uint' value"
    Expected: 130071681773528500889
```

```
      Actual: 13007168212868480739 3


Failure: test_s2_selectiveWithdraw_megaIndirectLowerToUpper_11DAI_74USDC_74USDT_from_

  "Error: Wrong `uint' value"
    Expected: 15914548952006536675 6
      Actual: 15914548948995620727 7


Running 30 tests for src/test/withdraws/views/suiteOneViews.t.sol:SelectiveWithdrawSu
[PASS] test_s1_viewSelectiveWithdraw_fullUpperAntiSlippage_5DAI_2SUSD_from_145DAI_90U
[PASS] test_s1_viewSelectiveWithdraw_partialLowerIndirectAntiSlippage_40DAI_40USDT_fr
[PASS] test_s1_viewSelectiveWithdraw_megaIndirectWithdrawLowerToUpper_11DAI_74USDC_74
[OOPS] testFailSelectiveWithdraw_lowerHaltCheck30Pct()
[FAIL] test_s1_viewSelectiveWithdraw_fullUpperAntiSlippage_5CDAI_2ASUSD_from_145DAI_9
[PASS] test_s1_viewSelectiveWithdraw_fullIndirectLowerAntiSlippage_5DAI_5USDC_from_95
[PASS] test_s1_viewSelectiveWithdraw_megaUpperToLower_95USDT_35SUSD_from_90DAI_90USDC
[FAIL] test_s1_viewSelectiveWithdraw_partialUpperAntiSlippage_50CUSDC_18SUSD_from_90D
[PASS] test_s1_viewSelectiveWithdraw_partialLowerSlippage_3DAI_60USDC_30USDT_1SUSD_fr
[PASS] test_s1_viewSelectiveWithdraw_fullLowerSlippage_1USDC_7USDT_2SUSD_from_95DAI_9
[PASS] test_s1_viewSelectiveWithdraw_smartHalt_lower_outOfBounds_to_inBounds() (gas:
[PASS] test_s1_viewSelectiveWithdraw_partialLowerSlippage_balanced_5DAI_5USDC_47USDT_
[PASS] test_s1_viewSelectiveWithdraw_smartHalt_lower_outOfBounds_to_outOfBounds() (ga
[PASS] test_s1_viewSelectiveWithdraw_partialUpperAntiSlippage_50USDC_18SUSD_from_90DA
[FAIL] test_s1_viewSelectiveWithdraw_fullIndirectLowerAntiSlippage_5CHAI_5CUSDC_from_
[PASS] test_s1_viewSelectiveWithdraw_fullUpperSlippage_8DAI_2USDC_8USDT_2SUSD_from_90
[OOPS] testFailSelectiveWithdraw_upperHaltCheck10Pct()
[PASS] test_s1_viewSelectiveWithdraw_fullLowerAntiSlippageWithdraw_5DAI_5USDC_0p5USDT
[PASS] test_s1_viewSelectiveWithdraw_smartHalt_lower_outOfBounds_exacerbated() (gas:
[PASS] test_s1_viewSelectiveWithdraw_smartHalt_upper_outOfBounds_to_inBounds() (gas:
[OOPS] testFailSelectiveWithdraw_lowerHaltCheck10Pct()
[PASS] test_s1_viewSelectiveWithdraw_smartHalt_upper_unrelated() (gas: 346355)
[PASS] test_s1_viewSelectiveWithdraw_partialLowerAntiSlippage_0p0001DAI_41USDC_41USDT
[PASS] test_s1_viewSelectiveWithdraw_balanced_10DAI_10USDC_10USDT_2p5SUSD_from_300Prc
[PASS] test_s1_viewSelectiveWithdraw_lightlyUnbalanced_5DAI_1USDC_3USDT_1SUSD_from_80
[PASS] test_s1_viewSelectiveWithdraw_fullIndirectUpperSlippage_5DAI_5USDT_from90DAI_1
[OOPS] testFailSelectiveWithdraw_upperHaltCheck30Pct()
[PASS] test_s1_viewSelectiveWithdraw_megaIndirectLowerToUpper_11DAI_74USDC_74USDT_frc
[PASS] test_s1_viewSelectiveWithdraw_partialUpperSlippage_balanced_0p001DAI_40USDC_40
[PASS] test_s1_viewSelectiveWithdraw_smartHalt_upper_outOfBounds_to_outOfBounds() (ga

VM error for testFailSelectiveWithdraw_lowerHaltCheck30Pct()
Failure: test_s1_viewSelectiveWithdraw_fullUpperAntiSlippage_5CDAI_2ASUSD_from_145DAI

  "Error: Wrong `uint' value"
    Expected: 6994286984194756641
      Actual: 6996036011379633519


Failure: test_s1_viewSelectiveWithdraw_partialUpperAntiSlippage_50CUSDC_18SUSD_from_9
```

```
  "Error: Wrong `uint' value"
    Expected: 67991384639438932784
      Actual: 68008385735861111167


Failure: test_s1_viewSelectiveWithdraw_fullIndirectLowerAntiSlippage_5CHAI_5CUSDC_fro

  "Error: Wrong `uint' value"
    Expected: 9992948093387737702
      Actual: 9995445955431676811


VM error for testFailSelectiveWithdraw_upperHaltCheck10Pct()
VM error for testFailSelectiveWithdraw_lowerHaltCheck10Pct()
VM error for testFailSelectiveWithdraw_upperHaltCheck30Pct()
```

# Appendix 3 - Disclosure

ConsenSys Diligence ("CD") typically receives compensation from one or more clients (the "Clients") for performing the analysis contained in these reports (the "Reports"). The Reports may be distributed through other means, including via ConsenSys publications and other distributions.

The Reports are not an endorsement or indictment of any particular project or team, and the Reports do not guarantee the security of any particular project. This Report does not consider, and should not be interpreted as considering or having any bearing on, the potential economics of a token, token sale or any other product, service or other asset. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. No Report provides any warranty or representation to any Third-Party in any respect, including regarding the bugfree nature of code, the business model or proprietors of any such business model, and the legal compliance of any such business. No third party should rely on the Reports in any way, including for the purpose of making any decisions to buy or sell any token, product, service or other asset. Specifically, for the avoidance of doubt, this Report does not constitute investment advice, is not intended to be relied upon as investment advice, is not an endorsement of this project or team, and it is not a guarantee as to the absolute security of the project. CD owes no duty to any Third-Party by virtue of publishing these Reports.

PURPOSE OF REPORTS The Reports and the analysis described therein are created solely for Clients and published with their consent. The scope of our review is limited to a review of Solidity code and only the Solidity code we note as being within the

scope of our review within this report. The Solidity language itself remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond Solidity that could present security risks. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty.

CD makes the Reports available to parties other than the Clients (i.e., “third parties”) – on its website. CD hopes that by making these analyses publicly available, it can help the blockchain ecosystem develop technical best practices in this rapidly evolving area of innovation.

LINKS TO OTHER WEB SITES FROM THIS WEB SITE You may, through hypertext or other computer links, gain access to web sites operated by persons other than ConsenSys and CD. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites’ owners. You agree that ConsenSys and CD are not responsible for the content or operation of such Web sites, and that ConsenSys and CD 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 ConsenSys and CD 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 Reports. ConsenSys and CD assumes no responsibility for the use of third party software on the Web Site and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

TIMELINESS OF CONTENT The content contained in the Reports is current as of the date appearing on the Report and is subject to change without notice. Unless indicated otherwise, by ConsenSys and CD.

CONTACT US



## Subscribe to Our Newsletter

Stay up-to-date on our latest offerings, tools, and the world of blockchain security.

e-mail address