

# 1 Sliding Tax Mechanism

## 1.1 Introduction

The MOMOToken introduces a unique taxation system that has been engineered to incentivize early and frequent trading on the Uniswap V2 Pair. This system is designed to balance between rewarding early adopters and ensuring a sustainable token economy in the long run.

## 1.2 Dynamic Taxation

Upon the inception of trading, the tax rate begins at 20%. With each subsequent purchase transaction, this tax rate is decremented by 1, all the way down to a static 1%. This sliding scale mechanism encourages buyers to trade earlier, as their trades will be taxed at a lesser rate over time.

$$\text{taxAmount (for purchase)} = \left( \frac{\text{taxRate (initially 20\% and decrementing)} \times \text{amount}}{100} \right) \quad (1)$$

For sellers, the tax mechanism is slightly different. Instead of accumulating tax, a portion of the tokens they sell is burned based on the current tax rate, effectively reducing the total supply and potentially increasing the token's scarcity and value.

$$\text{burnAmount (for sale)} = \left( \frac{\text{taxRate} \times \text{amount}}{100} \right) \quad (2)$$

## 1.3 Accumulation and Redistribution

Taxes accumulated from buyers are not immediately redistributed. They are accumulated in a pool, and once the accumulated tax exceeds a predefined threshold (`redistributionThreshold`), it is eligible for redistribution.

$$\text{accumulatedTax} + = \text{taxAmount} \quad (3)$$

## 1.4 Caller Reward Mechanism

To ensure that the tax redistribution and liquidity addition processes are executed regularly, the system rewards the caller (the person who triggers the redistribution). When the accumulated tax meets the redistribution threshold, half of it is utilized for liquidity and rewarding the caller:

$$\text{halfTax} = \frac{\text{accumulatedTax}}{2} \quad (4)$$

The caller's reward is 4.2% of the half tax:

$$\text{rewardForCaller} = \left( \frac{\text{halfTax} \times 42}{1000} \right) \quad (5)$$

The remaining amount, after rewarding the caller, is swapped for ETH and then used to add liquidity to the Uniswap pair. This mechanism ensures that the tax redistribution and liquidity provision processes are carried out regularly without relying on centralized intervention, thereby enhancing the decentralization aspect of the token's economy.

## 1.5 Conclusion

The MOMOToken's sliding tax mechanism, combined with the caller reward system, has been devised to foster a healthy token ecosystem. By incentivizing early and frequent trading and ensuring regular liquidity provision, the token aims to achieve both short-term trading activity and long-term sustainability.

# 2 Vesting Mechanism

## 2.1 Introduction

The MOMOToken introduces a vesting mechanism, often referred to as "reward claiming", which is designed to encourage long-term holding of the token. This mechanism ensures that holders can claim rewards periodically based on their token holdings, reinforcing the value proposition of holding onto the token.

## 2.2 Claim Interval and Cliff

Before a holder can claim any rewards, they must wait for a specific duration known as the "cliff". After this period, they can start claiming rewards at predefined intervals.

- **Cliff Duration:** `CLIFF_DURATION` = 30 days
- **Claim Interval:** `CLAIM_INTERVAL` = 420 hours

Thus, a holder can only claim rewards after:

$$\text{lastClaimTime} + \text{CLAIM\_INTERVAL} + \text{CLIFF\_DURATION} \quad (6)$$

## 2.3 Reward Calculation

The reward a holder can claim is proportional to their token holdings relative to the total supply and the reward pool. The formula for calculating the reward for a holder is:

$$\text{reward} = \left( \frac{\text{balanceOf(holder)}}{\text{totalSupply}} \right) \times \text{rewardPool} \quad (7)$$

Where:

- `balanceOf(holder)` represents the number of tokens held by the holder.
- `totalSupply` is the total supply of MOMOTokens.
- `rewardPool` is the accumulated pool of tokens reserved for rewards.

Once the reward is claimed, the holder's last claim timestamp is updated, and they have to wait for another claim interval after the cliff duration to claim rewards again.

## 2.4 Implications

The vesting mechanism ensures a steady and predictable token release schedule. By tying the release of rewards to the amount of token held and the total supply, it ensures fairness and proportionality. Moreover, the cliff and claim interval deter short-term speculative trading and encourage genuine, long-term holding, which can lead to a more stable token value.

article amsmath,amsfonts,amssymb graphicx hyperref [utf8]inputenc

# 3 Additional Mechanisms and Features

## 3.1 Anti-bot Measures

The MOMOToken contract is designed to mitigate the effects of trading bots, which often disrupt fair trading. The cooldown mechanism is employed to prevent rapid, successive transactions. After purchasing the token, a user (or bot) has to wait for a predefined duration before making another purchase. This cooldown effectively deters trading bots that rely on high-frequency trades.

$$\text{cooldownDuration} = 60 \text{ seconds} \quad (8)$$

## 3.2 Liquidity Redirection

To ensure a robust trading environment, the contract redirects a portion of the accumulated tax to the liquidity pool on Uniswap V2. This redirection not only ensures a healthy liquidity but also aims to minimize price volatility.

Given the accumulated tax, half of it is utilized for liquidity:

$$\text{liquidityAddition} = \frac{\text{accumulatedTax}}{2} \quad (9)$$

## 3.3 Reentrancy Defense

The contract utilizes the 'ReentrancyGuard' to protect against nested or recursive calls, a common vulnerability in smart contracts. This mechanism ensures that a function cannot be re-entered before it finishes execution, safeguarding against potential attacks.

### 3.4 Token Fundamentals

The total supply of the MOMOToken is given by:

$$\text{totalSupply} = 420 \times 10^{12} \times 10^{18} \quad (10)$$

This represents 420 trillion tokens, each with 18 decimals, adhering to the standard ERC20 precision.

### 3.5 Liquidity Pool Creation and Management

To facilitate trading, the contract interacts with the Uniswap V2 platform. When the pair address is set, the contract fetches the liquidity pair for the token and Ethereum (WETH) from Uniswap:

$$\text{uniswapV2Pair} = \text{getPair}(\text{address}(\text{this}), \text{uniswapV2Router.WETH}()) \quad (11)$$

This pair address is pivotal for trading and liquidity provision on the Uniswap platform.