LISTEDRESERVE

Part 2: Transactions and the Bitcoin Money Supply

September 2019

Research Note from ListedReserve

This short series, which does not represent investment advice, sets out to explain Bitcoin. In some cases, we will go into detail to explain some of the genius incorporated into the little-understood protocol. It remains to be seen if Bitcoin is a success, but in reading this series, you will realise that some of the great discoveries in modern mathematics and computer science are at work here.

In this note, we cover the Bitcoin money supply, how it interacts with mining and how we spend and receive bitcoins.

---

1. Introduction

In part 1, we covered the tools that Bitcoin uses—cryptography, elliptic curves, hashing, Merkle roots—how these are all used to generate proof of work. We will now consider the Bitcoin money supply and how it is restricted. We will discuss transactions, how they are constructed and sent and a specific type of transactions known as Coinbase transactions.

2. The Bitcoin Money Supply

We covered Bitcoin mining in part 1. Mining is crucial to the ecosystem for both its security and the money supply. New bitcoin enters the ecosystem through Bitcoin mining. Miners issue new coins to themselves when they mine a block, and these new coins are called Coinbase transactions.

The Bitcoin Money Supply

There will only ever be 21 million bitcoins. The current supply is 17.5 million.

Every ten minutes, 12.5 new coins are issue to a miner. That will drop to 6.25 coins in May 2020, 3.125 coins in 2024 and so on until there are 21 million coins in 2040.

With the help of some definitions embedded in the Bitcoin codebase, we will see how the Bitcoin money supply works:

- **COIN** is a constant and represents 100,000,000 satoshis (one bitcoin).
- **nSubsidy** is the block reward given to miners for mining a block.
- **nfees** are the transactions fees associated with a Bitcoin block.
- **nheight** is the number of blocks that have been mined up to this point.
- **SubsidyHalvingInterval** is a constant: 210,000. The Bitcoin reward halves every 210,000 blocks (about four years).

The code behind the money supply looks like this:

```
{
        nSubsidy = 50 x COIN
            halvings = nheight/subsidy halving interval
            nSubsidy >>= halvings                    [>> is a function in C++ that divides by 2]
            if (halvings >= 64) return nFees
            return nSubsidy + nFees
            }
```

In English this means

.

1. The base block reward is 50 BTC.
2. We then calculate how many halvings we have had by referencing the current block height and the subsidy halving interval. We are currently at block 584,877/210,000 = 2 halvenings
3. The code >>= is known as the binary right shift operator. It simply divides by two for each halving calculated above. In our example, it would be 50 ÷ 4 = 12.5 BTC—the current block reward in 2019.
4. Once we go past 64 halvings, the block reward becomes zero, and the miner only gets fees.
5. Otherwise, Bitcoin rewards the miner with the subsidy in 3 and the fees.

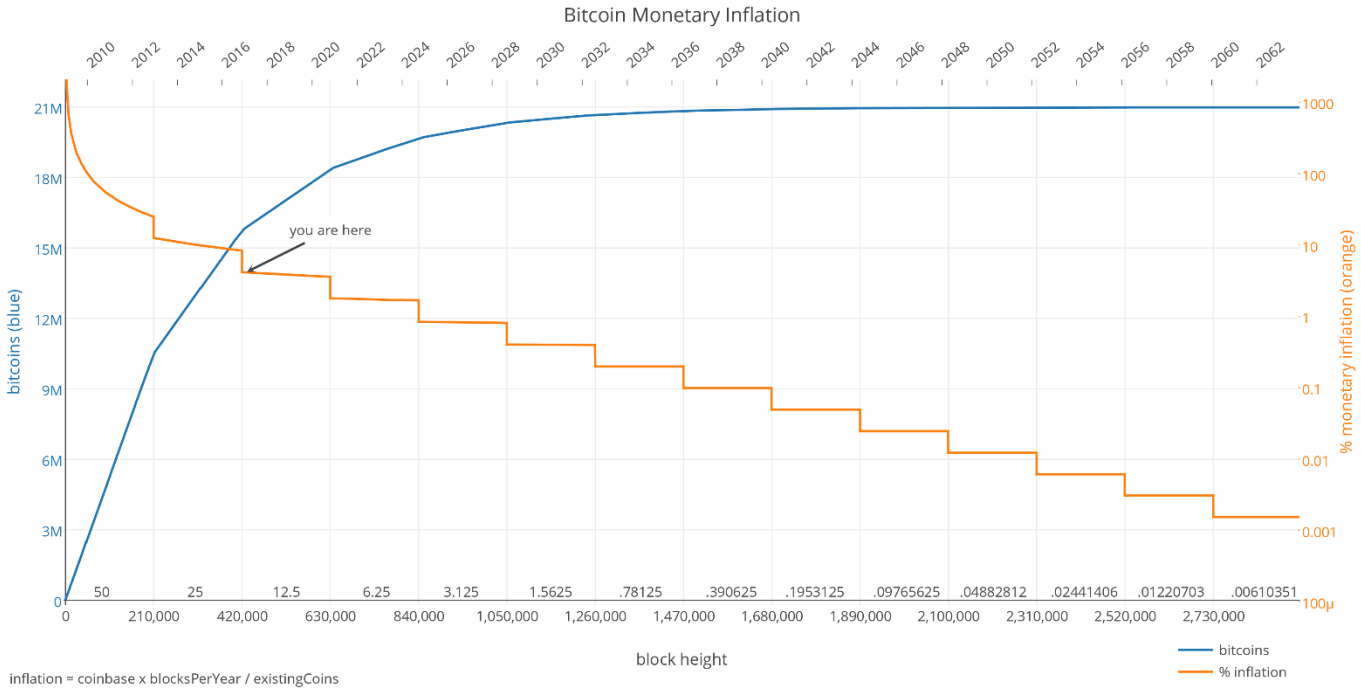
Even if you are not computer literate, if you spend a few minutes with the code and definitions above, you will understand it. The entire Bitcoin money supply is defined in five lines of code. Even better, check the code yourself; it is public. Click the link and search for 'nSubsidy = 50 x COIN':

https://github.com/bitcoin/bitcoin/blob/master/src/validation.cpp

This system stands in stark contrast to central banking, which is so complex that hardly anyone understands it. It is impossible to verify the money supply of EUR, USD, or any other major currency. In Bitcoin, the money supply is simple and known, and you can check it yourself.

3. The Supply Curve

The code defines the money supply. It starts with 50 bitcoins and grows by 50 bitcoins every ten minutes. After four years, this cycle drops to 25 bitcoins every ten minutes. In 2019, it is 12.5 bitcoins every ten minutes. By 2030, 99% of all bitcoins will be mined. By that point, inflation will be incredibly low at around 0.1% per annum.



4. Coinbase Transactions

Coinbase Transactions
The first transaction in every block is the block reward paid to the miner. These transactions are known as Coinbase transactions. Coinbase transactions generate the money supply.

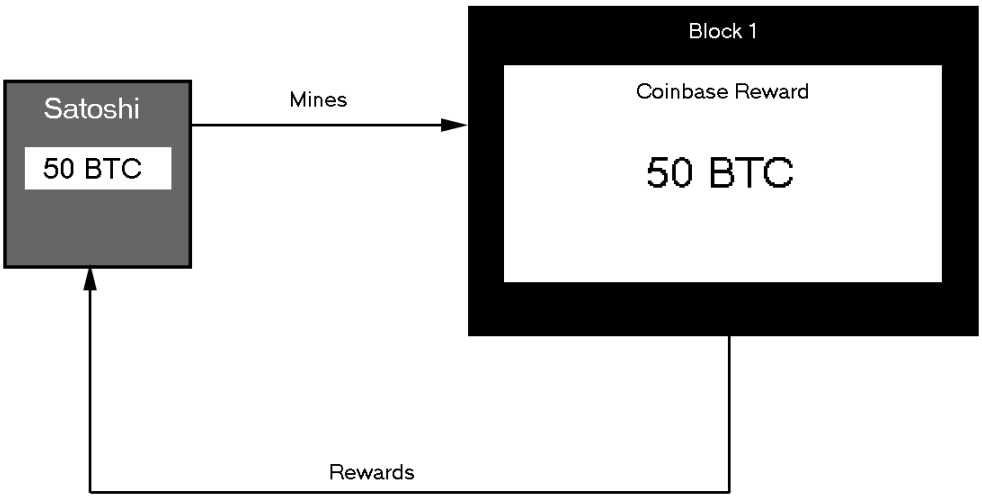
Instead of a central bank controlling the money supply through the banking system, Bitcoin's money supply is administered through the Coinbase transactions contained in each block. These Coinbase transactions are defined by the code in section 2 above.

Every block mined in the blockchain has a Coinbase transaction as its first transaction. These transactions contain the payment to the miner that mined that particular block and are comprised of

1. **The Block Reward**: The block reward is a newly minted amount of bitcoin algorithmically determined by the code above whereby the number of blocks already mined determines the amount to mint and then reward to the miner.
2. **Transaction Fees**: The miner also collects the transactions fees of the transactions they have selected for inclusion in the block—generally the transactions with the highest fees first.

Block 1: The First Bitcoin Block
The first-ever Bitcoin block simply mined 50 BTC and awarded it to Satoshi. The total supply at this point is 50 BTC with only one block.

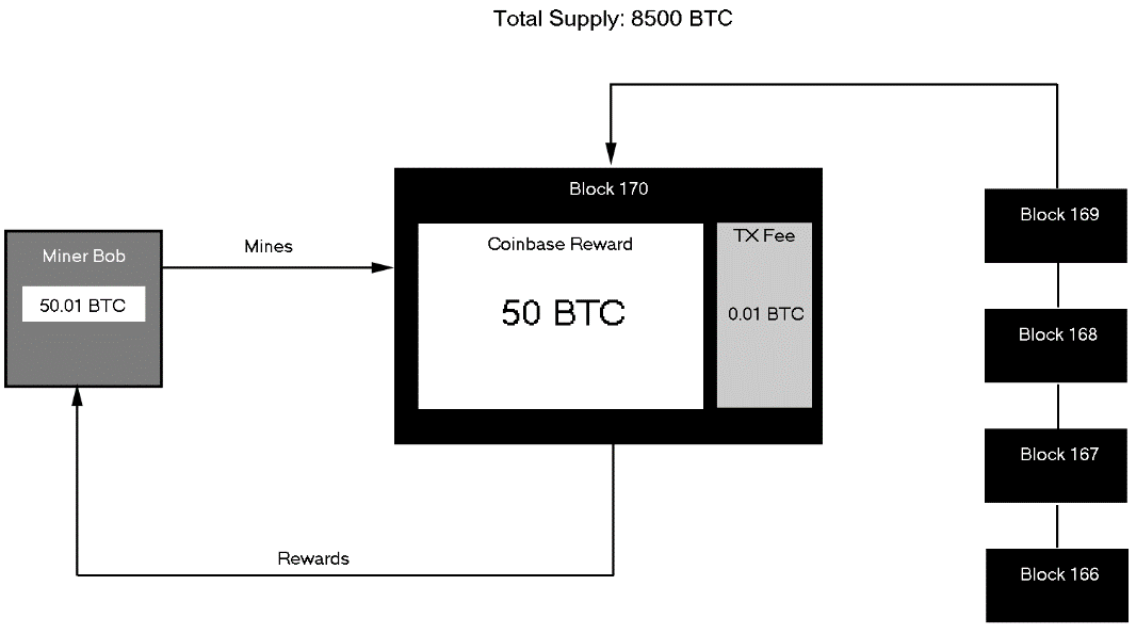
Total Supply: 50 BTC



The genesis block—or Block 1—was the first block mined into the blockchain. Its initial block reward of 50 BTC was paid via a Coinbase transaction to one of Satoshi's addresses, minting the first 50 bitcoin in existence. There were no transactions beyond this Coinbase transaction because they were no other bitcoins in existence; thus, there were no transaction fees.

There is a huge incentive for miners to participate here—they receive a major portion of the currency base for minimal work.

Block 170
Moving forward to Block170, we find that Bitcoin is now a few days old. The total supply is now 170 x 50 = 8,500. The miner here receives the 50 BTC reward plus 0.01 BTC.



Block 170 is a significant block because it contained the first transaction in the protocol that wasn't a Coinbase transaction. It was also the first time the Coinbase transaction contained more than just the block reward; miners were paid a minimal fee for the single peer-to-peer transaction that was included in the block.

This process continues with each block containing the 50 BTC reward and then all the transactions in the block.

Where We Are Now

As of 17 July 2019, we are at block 585,779. There are 17.8 million bitcoins in circulation (85% of total supply), and the block reward is 12.5 BTC. On average, mining fees are about 1 BTC per block.

Over time, transaction fees have risen as a percentage of the overall reward. By 2040, they will represent the entire reward.

5. Why Miners Don't Simply Add More to Their Coinbase Transactions

Cheating

If miners try to cheat and award additional bitcoin to themselves, they must publish that fact when they solve the block and send it across the network.

They know the nodes will reject the block and cost them about USD 150,000.

The incentive mechanism prevents anyone from cheating in this way, but people still try (and always fail).

In part 1, we noted that miners must create a candidate block for submission to the network with the following components:

i) Software version number
ii) Previous block's hash
iii) Merkle root of the selected transactions (a hash of the root of the transactions selected for the block)
iv) Timestamp (seconds from the Unix epoch)
v) Difficulty target (proof-of-work difficulty target for this block)
vi) Nonce

With the Merkle root, the miner must summarise all the transactions the miner is looking to mine. The Coinbase transaction is always the first transaction in the Merkle tree. If the miner is successful, he will publish his block containing all this data to the network.

Upon publishing a block, the rest of the network checks the transactions, including the Coinbase transaction, to make sure they are valid and comply with the rules. Specifically, the functions in the codebase are 'CheckBlock' and 'CheckBlockHeader'.

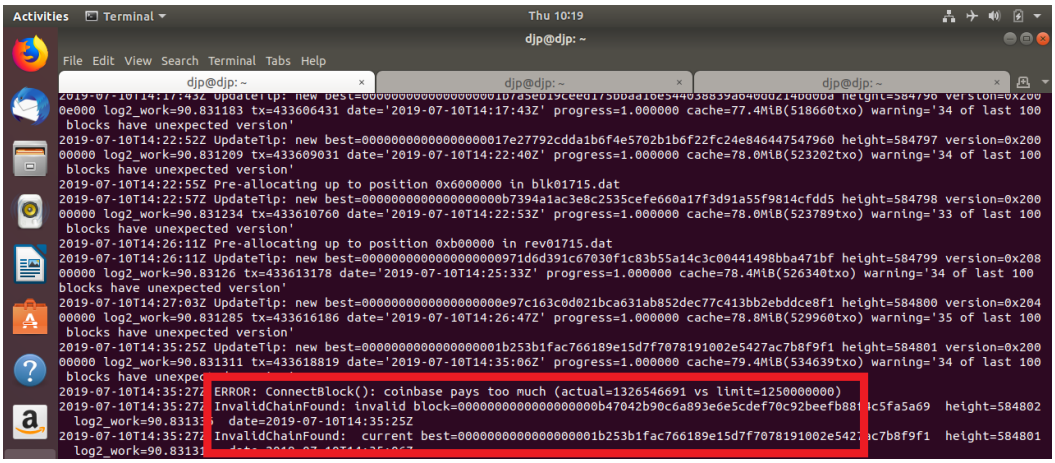
6. If a Miner Tries to Cheat

Let's look at a recent example of what happens when a miner tries to cheat.

On 10 July 2019 at block height 584,801, a miner tried to award himself 13.26 BTC, not 12.5 BTC, and our node at ListedReserve rejected the block (as did all others did across the world). See the error message 'Coinbase pays too much' in the image below. The miner.

This action is distributed consensus in action—everyone is monitoring the blockchain with a node. When someone breaks the rules of the protocol, it is obvious. Their block becomes invalid, and they lose the reward. In this case, the miner lost USD 150,000-worth of Bitcoin by trying to cheat.*

Cheating hardly ever happens because nobody wants to risk throwing away the block reward and fees.



** What happened in this case was that the block was not perfectly formed. The 13.26 BTC was 12.5 BTC plus the fees from the transactions. This reward correct, but the miner forgot to include the transactions generating the fees in the Merkle tree. Therefore, the block was rejected.*

7. Bitcoin Transactions

What Is a Bitcoin?
Bitcoins are transactions. Every bitcoin is the output of another Bitcoin transaction.

The basis of Bitcoin's transactions is the system of inputs and outputs. There are no accounts or balances in Bitcoin; there are only unspent transaction outputs (UTXOs). Anytime an address is credited with owning bitcoin, the balance simply refers to the amount of UTXOs that belong to that particular address.

Here's how it works:

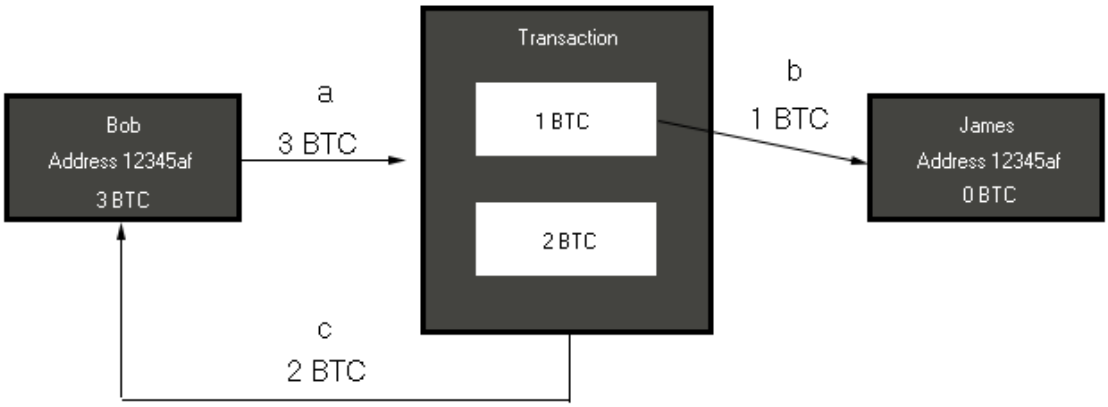
Imagine the first-ever transaction that was mined. 50 BTC, a Coinbase transaction, now belongs to Satoshi. Let's call it $UTXO_1$. $UTXO_1$ is not a balance. It is a transaction containing 50 BTC. Satoshi sends 30 BTC to his friend Hal. This creates

- $UTXO_2$ belonging to Satoshi with 20 BTC
- $UTXO_3$ belonging to Hal with 30 BTC

Every transaction creates new UTXOs. These transactions will be mined in the next block, along with a new Coinbase transaction.

Let's look at another example:

Bob has 3 BTC from a single transaction output. Bob wants to send 1 BTC to James.



Bob must use his entire 3 BTC UTXO as the input (a) for the transaction. The transaction then assigns the output of 1 BTC to James's address (b) and 2 BTC back to Bob's address (c). The UTXOs here (shown in the transaction box) are the amounts 'sent' to each address by the transaction. Bob now has a single UTXO of 2 BTC, while James has a single UTXO of 1 BTC.

This arrangement is the most abstract part of the UTXO system. When you make a transaction, you are utilising the entire amount or your UTXO (you can't split it), and the 'change' you get from the transaction goes to a new UTXO.

With UTXOs constantly being split, merged and manipulated to match the requirements of new transaction inputs, it may seem as though this system is excessively complex. However, Bitcoin wallets do all this work for us. Users simply see the sum of all their UTXOs in their wallets and assume they are seeing a balance when what they are seeing is a summation of transaction outputs.

The amount of bitcoins in circulation at any point in time is simply the sum of all UTXOs.

Bitcoin Wallets
Your Bitcoin wallet's 'balance' is just adding up your UTXOs and organising them in a way that is easy for you to visualise and use.

8. Public Keys, Private Keys and Bitcoin Addresses

Now we will look at how you keep your UTXOs (your bitcoins) secure. All the tools we need to do so have been covered in part 1. Note that a public key is not a Bitcoin address, and a Bitcoin address is not a UTXO. They are all distinct.

Security, Private Keys and Public Keys

Bitcoins (UTXOs) are secured by encryption.

All bitcoins have a private and public key associated with them. Unless you have the private key to a Bitcoin address, you cannot spend them.

There is a 1 in 2^{256} chance that someone could guess the private key. The largest computer in the world could not guess this key even if it began trying at the beginning of the universe.

Bitcoin is **very secure** and makes traditional systems look very insecure.

The Private Key (k)

Private keys are your keys to the Bitcoin kingdom. In Bitcoin, private keys are identified by a lowercase k. They protect your UTXO. Never, ever share them with anybody else under any circumstances.

Private keys must be generated randomly. They simply need to be any 256-bit number that no one could guess between 1 and 2^{256}, which as we know, is a very large number. One way to create this key is with a pen and paper (so it never touches the internet) and two dice:

- Roll the dice.
- If you roll nine or less, write down the number.
- If you roll ten, write down the number zero.
- If you roll 11 or higher, write nothing.
- Repeat xx times
- When you have your number, convert it to hexadecimal format (64 characters long).

Using this method, I got this number:

```
4513148943280021525786150910166722441228176762109154710063663433725396433285157392663997834610926074484660258157237415508698774818957707828892086118849186472754206954720476631222331734190315137861508439738788940720541557273269869330813342837340088683866886
```

I then converted it to this 64-character hexadecimal number:

```
KzjsSVNiPGrvHUr5nYVDv5CKHnpQWUFXA9By5WyUHaYezHdSuyhH
```

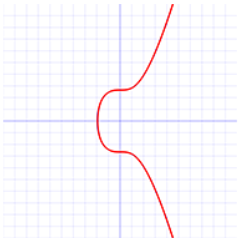
This your private key.

Public Key (K)

Recall cryptography from part 1:

Secp256k1 (Bitcoin) Elliptic Curve

$$y^2 = x^3 + 7$$



We used a common generator in our modular maths: the number 3. This system allowed both sides to generate the secret without knowing the other person's private number.

In Bitcoin, the common generator is larger. It is a pre-defined point on the elliptic curve above, and for completeness, it is known as G and looks like this:

```
02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798
```

It is always the same, and everyone in Bitcoin uses it. It is just a common point on the elliptic curve.

To generate your public key (K), we now generate another point on the elliptic curve by multiplying k by G.

Public Key = Private Key x G

K = k x G

Thanks to the properties we discussed in part 1, you cannot compute a private key from a public key. Even if you know K and G, you *cannot* compute k.

If I do that with the above, I get the following public key:

```
04DF517FF28B2E619A5D44363B913810FA82ED1A688295D3B8B5E371E25B240F47300B438FA8E0A59D66E9CFB2
3DDE492CD2B23FCD512F84ED840D8B11C08B742C
```

Bitcoin Addresses

The final part of the sequence is Bitcoin addresses. Addresses are hashed public keys and are just a fail-safe step to protect coins further. Bitcoin works with public keys as addresses, but these hashes move the reliance away from the elliptic curve alone.

Addresses are generated as follows:

1. The protocol takes my public key K and applies the SHA-256 hash (see part 1).
2. The protocol applies a further hash known as RIPEMD160, which generates shorter results than SHA, so they're easier to use.[1]
3. I receive a Bitcoin address that looks like this: `15yzoTvTKNcDcHUGf76ENKtgk1UzJCfETy`

Note how this is address shorter than what would be produced if we just used the public key. That's the only reason its hashed, and it's only hashed twice due to paranoia.

[1] The RIPE hash is used to keep they length of private keys down. Bitcoin would have worked by just using the RIPE hash alone and not the SHA hash first, but the creators were slightly worried because there is a relationship between the elliptic curve and RIPEMD160. While nobody knows if that relationship is useful or if they can prove anything meaningful with it, the Bitcoin creators didn't want to run the risk, and a simple solution was to apply SHA-256 first and break the suspected link between RIPE and ECC.

This step proves something: Bitcoin was written by people who didn't have all the answers. They knew there might be a slight weakness here in future (and I mean *slight* and the *distant* future), and they papered over it with a hash we probably don't need, just in case.

9. How Keys Protect Bitcoins

We have not covered how these keys actually protect your UTXOs.

Recall that UTXOs have two parts:

- An amount of bitcoin denominated in satoshis
- A locking script known as an encumbrance (specifies the conditions that need to be met to spend this coin)

In most cases, the locking script locks the output to a specific Bitcoin address. To unlock this script, you must have the private key.

As we have seen from the power of elliptic curve cryptography, nobody can compute your private key from your public key; nobody can ever guess it. In essence, that is the real power behind Bitcoin: property rights secured by mathematics, not by a social contract.

10. What's Next?

In part 3, we will cover an economic argument for Bitcoin:

- Scarcity and stock-to-flow ratios
- Central bank policy since 2008
- Government deficits
- The bond market
- Demography and its impact on digital adoption
- The retirement crisis
- Negative interest rates