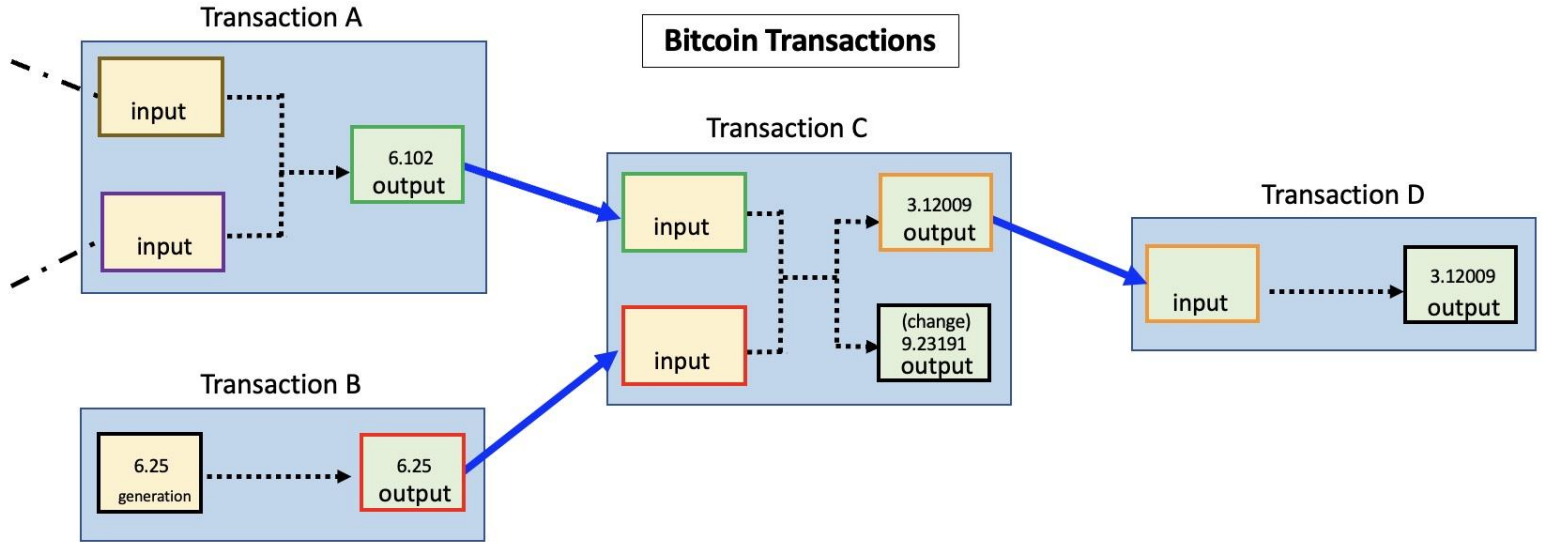
Bitcoin: Transactions, Blockchain, Mining

BLOCKCHAIN TECHNOLOGY

BITCOIN TRANSACTIONS

A Bitcoin transaction is simply the transfer of Bitcoin from one user to another. Think of it like sending digital cash.



Key elements:

- **Sender's address:** Like a bank account number, but for Bitcoin.
- **Recipient's address:** Where the Bitcoin is going.
- **Amount:** How much Bitcoin is being sent.
- **Digital signature:** A unique code that proves the sender authorized the transaction.

Once a Bitcoin transaction is confirmed and added to the blockchain, it is **extremely difficult, practically impossible, to reverse**.

TRANSACTION DATA STRUCTURE

Field	Size	Description
Version number	4 bytes	Used to specify rules to be used by the miners and nodes for transaction processing.
Input counter	1-9 bytes	The number (positive integer) of inputs included in the transaction.
List of inputs	Variable	Each input is composed of several fields, including `Previous Tx hash`, `Previous Txout-index`, `Txin-script length`, `Txin-script`, and optional sequence number. The first transaction in a block is also called a coinbase transaction. It specifies one or more transaction inputs.
Output counter	1-9 bytes	A positive integer representing the number of outputs.
List of outputs	Variable	Outputs included in the transaction.
Lock time	4 bytes	This field defines the earliest time when a transaction becomes valid. It is either a Unix timestamp or block height.

5. Null Data/OP_RETURN
Usage: Used to store small amounts of data on the blockchain (e.g., timestamping, proof of existence).
Uniqueness:
Non-redeemable and unspendable.
Helps prevent blockchain bloat while storing non-financial data.

TYPES OF TRANSACTIONS

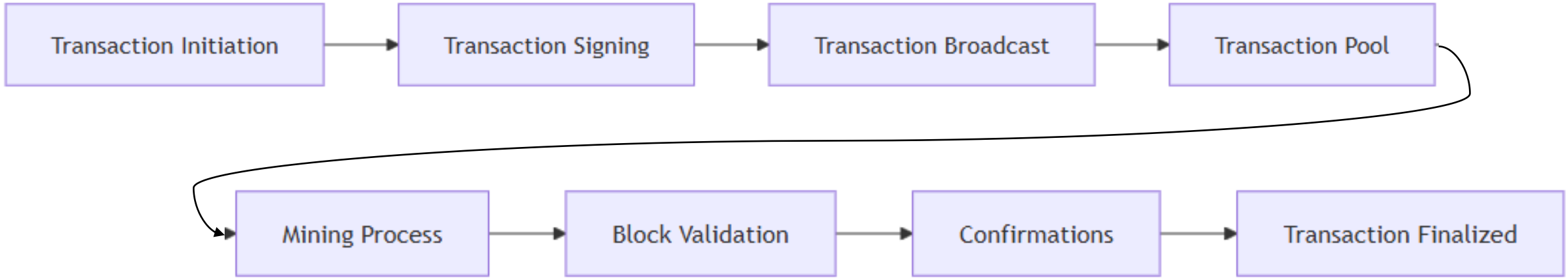
1. Pay to Public Key Hash (P2PKH)
- **Usage**: Commonly used for regular Bitcoin transactions.
- **Uniqueness**:
  - Simple, secure, and widely adopted.
  - Public key is hashed, improving security by hiding the actual public key.

2. Pay to Script Hash (P2SH)
- **Usage**: Used for complex transactions like multi-signature wallets, escrow, and custom conditions.
- **Uniqueness**:
  - Enables complex scripts to be executed on the Bitcoin network.
  - Flexibility to define custom spending rules (e.g., multi-sig).

3. MultiSig (Pay to MultiSig)
- **Usage**: Requires multiple signatures for authorization.
- **Uniqueness**:
  - Enhances security by requiring multiple parties to approve transactions.
  - Commonly used in joint accounts, escrow services, and corporate wallets.

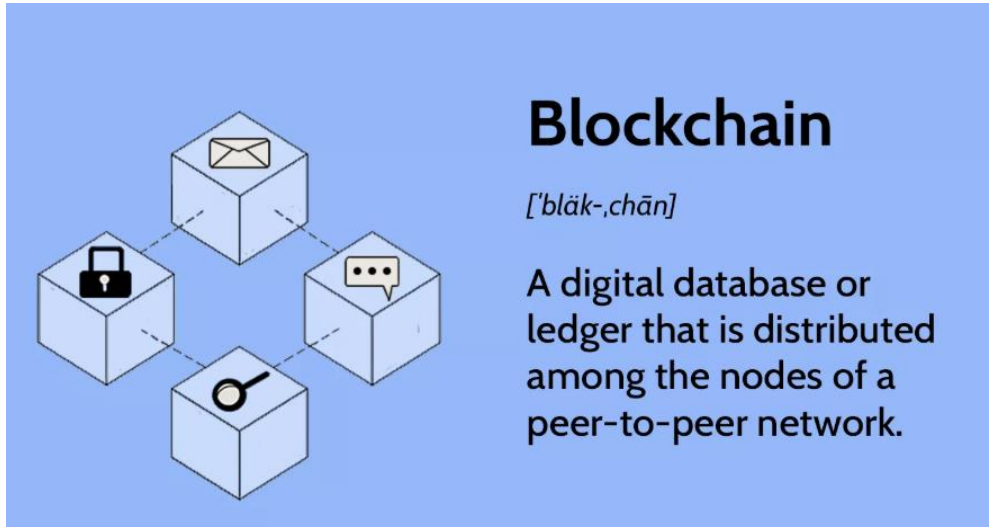
4. Pay to PubKey (P2PK)
- **Usage**: Used in early Bitcoin transactions, now largely obsolete.
- **Uniqueness**:
  - Public key is directly embedded in the transaction.
  - Obsolete due to security risks (exposing public keys).

Transaction LifeCycle



- **Transaction Initiation** – User/sender initiates a transaction via wallet software.
- **Transaction Signing** – The wallet signs the transaction using the sender's private key.
- **Transaction Broadcast** – The transaction is broadcasted to the Bitcoin network.
- **Transaction Pool (Mempool)** – The transaction enters the mempool, waiting for confirmation.
- **Mining Process** – Miners pick the transaction and attempt to solve the PoW problem.
- **Block Validation & Propagation** – The mined block is validated and propagated across the network.
- **Confirmations Begin** – Nodes verify the block and propagate confirmations.
- **Final Transaction Confirmation** – After sufficient confirmations (typically 3–6), the transaction is finalized.

Blockchain



The **blockchain** is a public, distributed ledger that records all Bitcoin transactions.

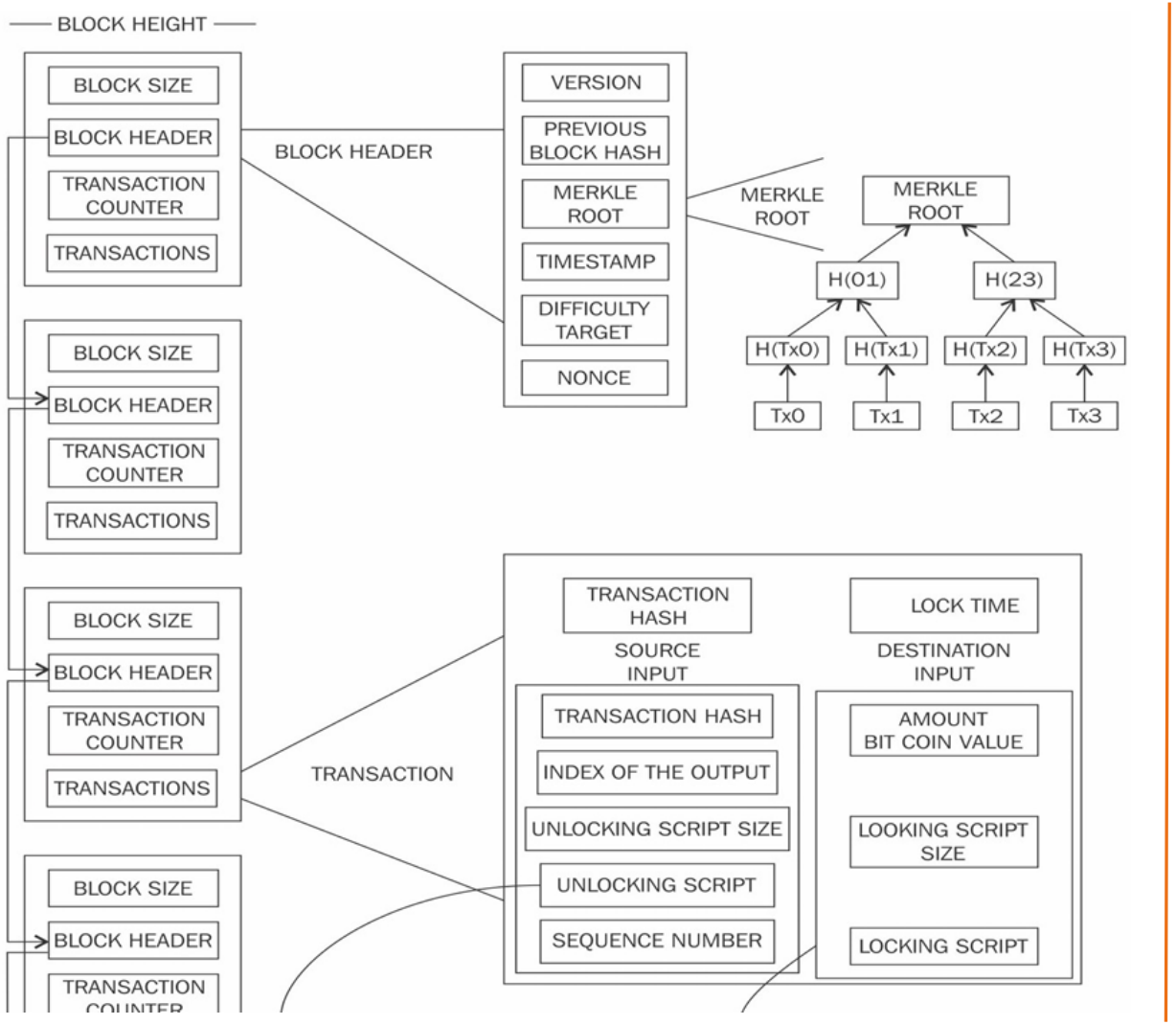
Key features:

- Immutable: Once a block is added, it cannot be changed.
- Decentralized: No single entity controls it.

How it works:

- Transactions are grouped into "blocks."
- Each block is linked to the previous one, forming a chain.
- The blockchain is maintained by nodes (computers) across the world.

Structure Of Block



1. Block Header (80 bytes)

- **Version (4 bytes)**: Indicates the software protocol version.
- **Previous Block Hash (32 bytes)**: Hash of the previous block.
- **Merkle Root (32 bytes)**: Root hash of the Merkle tree containing transactions.
- **Timestamp (4 bytes)**: UNIX timestamp of block creation.
- **Difficulty Target (4 bytes)**: Defines the difficulty of mining.
- **Nonce (4 bytes)**: Random number used for proof-of-work.
- **Transaction Counter (Varies)**: Number of transactions in the block.
- **Transactions (Varies)**: List of all transactions.

2. Merkle Tree Structure

- Used to efficiently verify transactions.
- Transactions are hashed recursively to form a single **Merkle Root**.
- **Merkle Root (32 bytes)**: Topmost hash summarizing all transactions.

3. Transaction Structure

- **Source (Input)**
  - **Transaction Hash (32 bytes)**: Hash of the previous transaction output being spent.
  - **Index of Output (4 bytes)**: Reference to the specific output being spent.
  - **Unlocking Script Size (Varies)**: Size of the unlocking script.
  - **Unlocking Script (Varies)**: Script that proves ownership of the input.
  - **Sequence Number (4 bytes)**: Used for time-lock transactions.
- **Destination (Output)**
  - **Amount Bitcoin Value (8 bytes)**: Amount of Bitcoin being transferred.
  - **Locking Script Size (Varies)**: Size of the locking script.
  - **Locking Script (Varies)**: Defines spending conditions for the output.

Mining

Mining is the process of adding new blocks to the blockchain and verifying transactions.

How mining works:

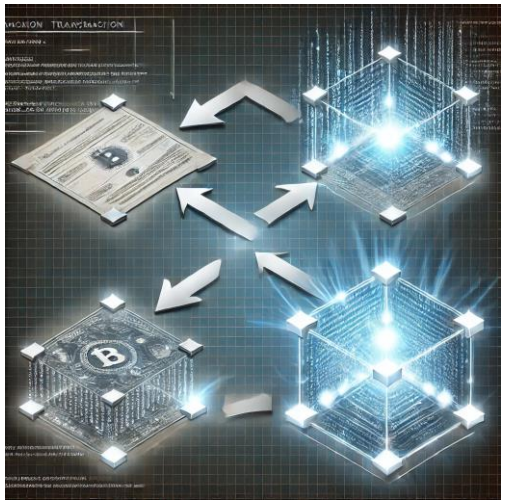
1. Miners compete to solve a mathematical puzzle (proof of work).
2. The winner adds the block to the blockchain.
3. The miner is rewarded with newly created Bitcoin.

Why is bitcoin mining important?

1.Network Security
2.Transaction Verification
3.Bitcoin Creation

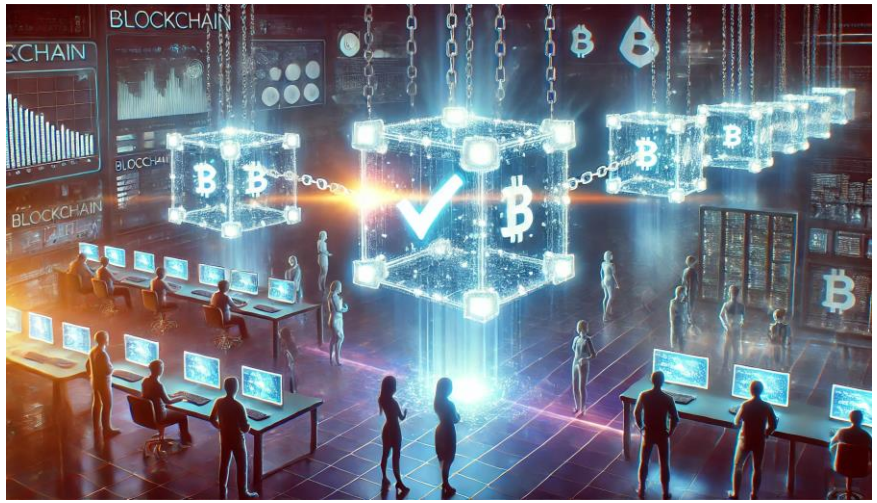
Transaction

Verification



Block creation

Illustration Of Mining



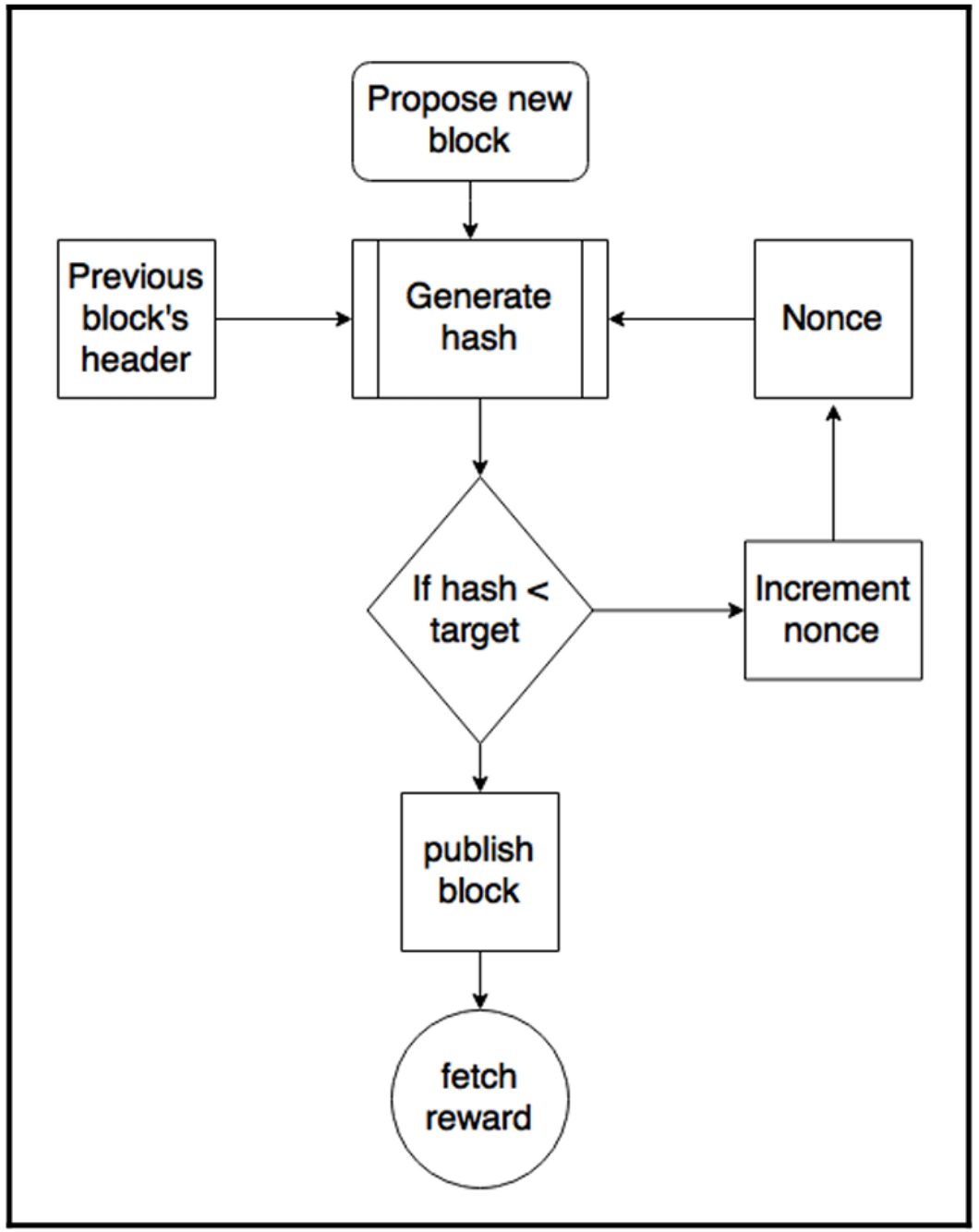
Blockchain update

Reward

Mining Algorithm

Mining Algorithm Steps:

- Retrieve the previous block's header
- Assemble a set of transactions to propose in a new block.
- Compute the double hash of the previous block’s header, combined with a nonce and the new block, using the SHA-256 algorithm.
- Check if the resultant hash is below the current difficulty level (target). If true, Proof of Work (PoW) is solved, and the block is broadcasted to the network for a reward.
- If the hash does not meet the target, increment the nonce and repeat the process.



THANK YOU

V. SUMANTH	22501A05I8
T. TEJ MAHENDRA	22501A05H9
SK. FAKRUDDIN	22501A05G1
P. KAMAL SIDDARDHA	22501A05D5
P. PHANI KRISHNA	22501A05D8