

\$76 / p76token

A memecoin minted via the p-token batch instruction

Whitepaper v2.0 · 14 May 2026

Abstract

\$76 is a memecoin on Solana. It exists as a tribute to one technical event: the activation of the p-token program upgrade on Solana mainnet on 13 May 2026. That upgrade cut the SPL Token Transfer cost from 4,645 compute units down to 76. It also introduced a new batch instruction with discriminator `0xFF`.

The token has a fixed supply of 76 whole tokens (76,000,000 atomic units at 6 decimals). No team allocation. No presale. No roadmap. Authorities revoked at genesis.

The genesis transaction itself uses the batch instruction wrapping a single `MintTo`, making \$76 the first memecoin minted via this code path on mainnet.

1. The moment

On 13 May 2026, the Solana network activated SIMD-0266. The legacy SPL Token program was replaced by the p-token implementation. Same on-chain address (`TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA`), complete rewrite using the Pinocchio framework.

The Transfer instruction (the most-called instruction on Solana) dropped from 4,645 CU to 76 CU. A 98.4% reduction. Every token transfer, every swap, every airdrop now consumes 76 compute units instead of 4,645.

In the same upgrade, a new instruction landed with discriminator `0xFF`. The batch instruction. It wraps multiple inner instructions in a single CPI, enabling atomic multi-action transactions without requiring custom programs.

\$76 is the first memecoin minted using this batch instruction.

2. Why a memecoin

Memecoins are how crypto cultures encode moments. Pepe was a frog meme. Wif had a hat. Bonk was a Solana dog.

\$76 is a compute unit count.

This is not a utility token. There is no DEX, no DAO, no governance, no staking, no protocol. There is a number (76) encoded across three independent parameters:

1. The new Transfer CU cost (76)

2. The total supply (76 tokens)
3. The ticker (\$76)

And there is a discriminator (0xFF) that defines how the token entered existence. Every \$76 traded carries the lineage of being minted by the same instruction that now powers a 98.4% efficiency gain on Solana.

That is the entire thesis. Hold it or do not.

3. Technical architecture

3.1 The batch instruction wire format

From `pinocchio/program/src/processor/batch.rs`:

```
pub const BATCH_IX_DISCRIMINATOR: u8 = 0xFF;

pub fn process_batch(
    accounts: &[AccountInfo],
    data: &[u8],
) -> ProgramResult {
    let mut offset = 0;
    while offset < data.len() {
        let n_accounts = data[offset];
        let ix_len = data[offset + 1];
        invoke_inner(accounts, &data[offset + 2..])?;
        offset += 2 + ix_len as usize;
    }
    Ok(())
}
```

The outer instruction data begins with byte 0xFF. After this byte, the data is a stream of inner instructions, each encoded as:

```
[num_accounts: u8][ix_data_length: u8][ix_data: bytes]
```

The outer accounts list is consumed `num_accounts` at a time per inner. There is no length prefix on the inner count. The loop runs until `data.len()` is exhausted. Each inner instruction's `ix_data` starts with its own discriminator byte (e.g. 0x07 for MintTo) followed by its payload.

3.2 The \$76 genesis batch payload

The exact 12-byte instruction data for the genesis transaction:

```
FF 03 09 07 00 AB 87 04 00 00 00 00
```

Decoded:

Byte(s)	Field	Value
FF	batch_discriminator	The new p-token entry
03	num_accounts	mint, dest_ata, mint_auth
09	inner_ix_data_length	9 bytes follow

07	MintTo discriminator	Standard SPL Token op
00 AB 87 04 00 00 00 00	amount (u64 LE)	76,000,000 atomic = 76 whole

On Solscan, this transaction parses as "Token Program: Batch" with an inner `MintTo` event for 76,000,000 raw units.

3.3 Why batch instead of legacy MintTo

A direct `MintTo` call would have been simpler. The batch wrapper adds 3 bytes of overhead (`FF`, `num_accounts`, `ix_data_length`) and a small amount of CPI overhead. We accept this cost because the entire token is a tribute to the batch instruction's existence. If we minted with the legacy code path, the token would be unremarkable. The batch wrapper is the point.

3.4 Compute unit footprint

The genesis batch transaction consumed 486 CU on mainnet. For context:

Operation	Legacy SPL	p-token
Transfer	4,645 CU	76 CU
MintTo	4,128 CU	2,012 CU
Genesis batch (1 MintTo)	n/a	486 CU

The 486 CU figure is recorded permanently in the on-chain transaction logs. It is the cost of bringing \$76 into existence.

4. Tokenomics

Parameter	Value
Total supply	76 (76,000,000 atomic)
Decimals	6
Mint authority	None (revoked at genesis)
Freeze authority	None (never set)
Team allocation	0%
Presale	None
Vesting	None
Initial liquidity	100% of supply in Meteora DAMM v2 pool

All 76 tokens were minted in one atomic transaction to a single ATA owned by the deployer wallet. Immediately after mint, the entire supply was deposited as single-sided liquidity into a Meteora pool with permanently locked liquidity. The deployer retains zero tokens.

5. The genesis ceremony

The on-chain birth of \$76 consists of five transactions, sent in this order from a single mainnet wallet:

TX 1: InitializeMint2

Creates the mint account with decimals = 6, mint authority = deployer wallet, freeze authority = null.

Signature:
43XGhq7QUHoYKzXYp5KpuFz7vCEMgFuTJV5G5NL3frjW
tYgJnYfcUkNVne2f9FLkouG4EXazRgv7awdyuQh566Lx

TX 2: CreateMetadataAccountV3

Creates the Metaplex Token Metadata account so the token displays name "p76token", symbol "76", and image on Solscan, Phantom, and aggregators. Update authority retained on deployer wallet.

Signature:
64n5GkeDsWPDjdRi4CwvfX421YnmJVLqBXYa1NQ7xqgj
htNA3MgbdG1muYZSD3nmt1srt2Vovgp7tv3DaepKreqG

Metadata URI:
<https://gateway.irys.xyz/>
5A1MBDfqUVMkDK6Ff2V6DheRchvaJBYcAEi7yeif9p2R

TX 3: CreatePayerATA

Creates the deployer's associated token account for the mint.

Signature:
4ngQcDV7Tih55snEP1pi4jj9gg9hYDRR8A7oqF6gCfez
2r3LgWJD9qJF6JYcwgZ4TSgyvzisQU5xKg7oKYN83rM1

TX 4: Genesis batch (THE TX)

A single batch instruction (0xFF) wrapping one inner MintTo of 76,000,000 atomic units (76 whole tokens) to the deployer ATA.

Signature:
5cqt7QpW2WhyQoJC7u8WhVivoUC5rWrHudWAssYrm2xD
TTvVszVAXyCidco5DAXqwc4u4DSAXmmt1kzCKwJP2iW8

Block: 419,735,580
Timestamp: 17:53:32 UTC, 14 May 2026
CU consumed: 486
Transaction size: 302 bytes
Instruction data: FF 03 09 07 00 AB 87 04 00 00 00 00

This is the canonical \$76 transaction. The `setComputeUnitPrice` and `setComputeUnitLimit` instructions precede the batch, as is standard for any mainnet transaction in 2026, but the third instruction is pure batch.

TX 5: SetAuthority MintTokens = None

Revokes the mint authority forever. After this transaction lands, no more \$76 tokens can ever be created. Supply locked at 76 permanently.

Signature:
38MWTJ1SuuHjb5hcU6ypZ5FXaGq4trxpwuHCihZGFjes
PQtegnjqUMxeyg2Y24SeSea6RzW2P2RzrZu6dzqGD4q

The freeze authority was never set, so no separate freeze revocation was needed.

6. Mathematics

6.1 Why 76, and only 76

Three reasons, in order of importance:

- 1. It matches the new CU cost.** The p-token Transfer instruction costs 76 CU. The supply mirrors this number.
- 2. It is small enough to be memorable.** Not 76 million, not 76 thousand. Just 76. Each token is a discrete unit with weight.
- 3. It produces a market structure where each \$76 represents roughly 1.3% of supply.** This is unusual for a memecoin. Most use 1 billion supply where each token is a rounding error. With 76, the unit matters.

6.2 Why decimals = 6

The 6 decimals come from SPL convention. We did not need decimals technically (76 indivisible tokens would have been the purest expression). Decimals = 6 allows micro-purchases at the cost of slightly diluting the "76 discrete tokens" narrative. We chose pragmatism over purity here. Buyers can hold fractional \$76, but the supply count remains 76.

6.3 Why batch wrapping a single MintTo

The batch instruction's narrative power requires using it for the genesis. Wrapping a single inner instruction is the minimum viable usage. It demonstrates the wire format without requiring distribution complexity. Future users of the batch instruction will write more interesting multi-inner sequences. \$76 is the first example, and it is deliberately minimal.

7. Tribute attribution

\$76 exists because four entities built the p-token upgrade:

Entity	Contribution
@solana (Solana Foundation)	Network, governance, SIMD process
@anza_xyz (Anza)	p-token implementation, Pinocchio framework
@toly (Anatoly Yakovenko)	Solana architecture
@0x_febo (Febo)	Original p-token POC, batch instruction design

This token carries no claim to be officially endorsed by any of these entities. The tribute is unidirectional. We minted because their work made this specific transaction shape possible. The token transfers cost 76 CU because of their work. The `0xFF` discriminator exists because of their work.

No royalties flow back to any of them. We chose not to wire creator fees into the metadata. The tribute is the existence of the token, not any cashflow.

8. Verification appendix

8.1 Contract addresses

Mint	7T1F8ERfwBgrArR8QMBQwlnaPv9YfBvDdrTdx3MBibb3
Deployer wallet	F3FjaJfcypligqKrQVaR8njASnt96xVEz2HFDAZx7umg
Deployer ATA	9HFNWeQkM6ekTHU57hpTdx7n7PA9usfkeGSUCNMiGVfX
Token Program	TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA
Metaplex Metadata	metaqbxxUerdq28cj1RbAWkYQm3ybzjb6a8bt518x1s

8.2 Transaction signatures

InitializeMint2	43XGhq7QUHoYKzXYp5KpuFz7vCEMgFuTJV5G5NL3frjW tYgJnYfcUkNVne2f9FLkouG4EXazRgv7awdyuQh566Lx
CreateMetadataAccountV3	64n5GkeDsWPDjdRi4CwvfX421YnmJVLqBXYa1NQ7xqgj htNA3MgbDGLmuYZSD3nmt1srt2Vovgp7tv3DaepKreqG
CreatePayerATA	4ngQcDV7Tih55snEP1pi4jj9gg9hYDRR8A7oqF6gCfez 2r3LgWJD9qJF6JYcwgZ4TSgyvzisQU5xKg7oKYN83rM1
Genesis batch	5cqt7QpW2WhyQoJC7u8WhVivoUC5rWrHudWAssYrm2xD TTvVszVAXyCidco5DAXqwc4u4DSAXmmt1kzCKwJP2iW8
Renounce mint auth	38MWTJ1SuuHjb5hcU6ypZ5FXaGq4trxpwuHCihZGFjes PQtEgjnjqUMxeyg2Y24SeSea6RzW2P2RzrZu6dzqGD4q

8.3 Independent verification commands

Anyone with a Solana CLI can verify the on-chain state:

```
# Verify mint state (supply, decimals, authorities)
solana account 7T1F8ERfwBgrArR8QMBQwlnaPv9YfBvDdrTdx3MBibb3 \
  --url https://api.mainnet-beta.solana.com

# Inspect the genesis batch transaction
solana confirm -v \
  5cqt7QpW2WhyQoJC7u8WhVivoUC5rWrHudWAssYrm2xDTTvVsz\
  VAXyCidco5DAXqwc4u4DSAXmmt1kzCKwJP2iW8 \
  --url https://api.mainnet-beta.solana.com
```

Expected output for the mint: supply 76,000,000 (= 76 whole at 6 decimals), mint authority None, freeze authority None.

Expected output for the genesis tx: status Finalized, instructions ComputeBudget setPrice, ComputeBudget setLimit, Token Program batch, inner instructions 1 MintTo of 76,000,000 to deployer ATA, CU consumed 486.

8.4 Block-height precedence claim

The genesis batch landed at slot 419,735,580 on 14 May 2026 at 17:53:32 UTC. This is approximately 29.4 hours after the p-token activation went live (13 May 2026, 12:39:34 UTC).

A search of mainnet transactions in the interval between activation and the \$76 genesis turned up no other token mint that uses the `0xFF` batch discriminator. The first known batch-instruction memecoin mint is therefore the \$76 genesis at block 419,735,580.

Anyone disputing this claim can search the block range using a Solana RPC and the discriminator filter. The data is public.

9. Disclosures

\$76 is a memecoin. It has no fundamental value, no cashflow, no protocol use, no revenue. Its only properties are the technical narrative encoded above and whatever market participants choose to pay for it.

Holders should expect that the token may trade at zero. The team retains zero allocation, so the team has zero ability to support a price floor. The liquidity in the Meteora pool is permanently locked, so the team cannot rug. But this does not prevent the price from going to zero on the open market.

Buying \$76 is a discretionary act of cultural participation in the p-token upgrade. Treat it as such, not as an investment.

10. Closing

The Solana network just got 98.4% cheaper for the most common operation it performs. That is a real thing that happened. Hundreds of thousands of transfers per day now cost 76 CU instead of 4,645 CU. The cumulative compute saved over a year of network operation is in the billions of units.

The work that made this happen deserves a marker. \$76 is that marker, encoded in the cheapest form that crypto culture has: a memecoin minted with the exact instruction whose existence we are celebrating.

If you trade \$76, you are participating in that marker. If you ignore it, that is also fine. The on-chain record stands either way.

Contract: `7T1F8ERfwBgrArR8QMBQw1naPv9YfBvDdrTdx3MBibb3`