# 24.12.31 Index Check Benchmark

Xun Zhang Wuyun Siqin Bingsheng Zhang
Zhejiang University, CHN
22221024@zju.edu.cn 3210101763@zju.edu.cn bingsheng@zju.edu.cn

December 31 2024

## 1 Relations and Circuit

The index check circuit is designed to prove the following relation:

- $\forall i : \mathsf{index}_i \leq m$ and $\forall i \neq j : \mathsf{index}_i \neq \mathsf{index}_j$.

Where $m$ is the total number of lottery. Each signer's lottery index should be less than total number, and there can not be two same lottery numbers of different signers. This index will be used in the eligibility check(as a input of mapping/hash function).

Our circuit implementation includes a compare gate for $\mathsf{index}_i$ and $m$, and also many gates for inequality check between every $\mathsf{index}_i$.

The code is like bellowing:

```
let less: AssignedValue<F> = big_less_than::assign(
    base_chip.range(),
    ctx,
    a: index.clone(),
    b: m.clone(),
    base_chip.limb_bits,
    limb_base: base_chip.limb_bases[1],
);

let equal: AssignedValue<F> = big_is_equal::assign(
    base_chip.gate(),
    ctx,
    a: index,
    b: m.clone(),
);

let result: AssignedValue<F> = base_chip.gate().or(ctx, a: less, b: equal);
```

This two gates is same as eligibility check gate, compare the index with $m$.

```
for i: usize in 0..len {
    for j: usize in i+1..len{
        let eq: AssignedValue<F> = base_chip.gate().is_equal(ctx, a: indexes_assigned[i], b: indexes_assigned[j]);
        let is_valid: AssignedValue<F> = base_chip.gate().is_zero(ctx, a: eq);
        is_valids.push(is_valid);
    }
}
```

This code compares each index pairwise, and constraint it to unequal.

## 2 Benchmark

The benchmark setting is $num\_limbs = 3$, and $limb\_bits = 90$.

Here is our index check benchmark result:

| Degree | Advice | Number | Proof Time | Proof Size | Verify Time |
|---|---|---|---|---|---|
| 18 | 1 | 128 | 5.2626s | 960 | 6.8613ms |
| 18 | 1 | 256 | 8.1488s | 1920 | 8.3849ms |
| 18 | 1 | 512 | 15.9302s | 4576 | 8.1831ms |
| 18 | 48 | 1024 | 44.1102s | 14944 | 12.5294ms |
| 20 | 48 | 2048 | 182.2723s | 14944 | 13.2670ms |

Table 1: Index Check Benchmark

Due to our circuit implementation, the proving cost of index check is $O(n^2)$, where $n$ is the number of indexes.

## 3 Parameters

We can find the parameters in the Mithril paper:

| | Adversarial Stake | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 40% | | | | 33% | | | |
| $\frac{k}{m}$ | $k$ | $m$ | L-Abs | L-Par | $k$ | $m$ | L-Abs | L-Par |
| $\phi(.55)$ | 2422 | 20973 | 99.999 % | $\approx 1$ | 856 | 7407 | $1-2^{-30}$ | $\approx 1$ |
| $\phi(.60)$ | 1445 | 11531 | 49.24 % | $\approx 1$ | 605 | 4824 | 99.667 % | $\approx 1$ |
| $\phi(.67)$ | 857 | 6172 | LL | $\approx 1$ | 414 | 2980 | 48.31 % | $1-2\cdot 10^{-18}$ |
| $\phi(.75)$ | 554 | 3597 | LL | $1-7\cdot 10^{-13}$ | 296 | 1921 | LL | $1-2\cdot 10^{-7}$ |
| $\phi(.80)$ | 445 | 2728 | LL | $1-5\cdot 10^{-7}$ | 250 | 1523 | LL | 99.98% |

**Table 2.** Required values of $k, n$ so that an adversarial quorum is formed with $P \leq 2^{-128}$. L-Abs and L-Par represent probability to form quorum (before retries) when the adversarial stake abstains or participates respectively. LL describes probabilities $< 1\%$. The parameters can be meaningfully used in conjunction with an incentive scheme or as an auxiliary opportunistic parametrization where a less aggressive parametrization is used as a fallback. Values of $\approx 1$ indicate a chance of failure $< 10^{-30}$.

The recommend parameters is around hundreds, so this part is not so expensive, although it is a super-linear proving task.