WIKIPEDIA
The Free Encyclopedia

Xoroshiro128+

xoroshiro128+ (named after its operations: XOR, rotate, shift, rotate) is a pseudorandom number generator intended as a successor to xorshift+. Instead of perpetuating Marsaglia's tradition of xorshift as a basic operation, xoroshiro128+ uses a shift/rotate-based linear transformation designed by Sebastiano Vigna in collaboration with David Blackman. The result is a significant improvement in speed and statistical quality.[1]

Statistical Quality

The lowest bits of the output generated by xoroshiro128+ have low quality. The authors of xoroshiro128+ acknowledge that it does not pass all statistical tests, stating

> This is xoroshiro128+ 1.0, our best and fastest small-state generator for floating-point numbers. We suggest to use its upper bits for floating-point generation, as it is slightly faster than xoroshiro128**. It passes all tests we are aware of except for the four lower bits, which might fail linearity tests (and just those), so if low linear complexity is not considered an issue (as it is usually the case) it can be used to generate 64-bit outputs, too; moreover, this generator has a very mild Hamming-weight dependency making our test (http://prng.di.unimi.it/hwd.php) fail after 5 TB of output; we believe this slight bias cannot affect any application. If you are concerned, use xoroshiro128** or xoshiro256+.
>
> We suggest to use a sign test to extract a random Boolean value, and right shifts to extract subsets of bits.
>
> The state must be seeded so that it is not everywhere zero. If you have a 64-bit seed, we suggest to seed a splitmix64 generator and use its output to fill s.
>
> NOTE: the parameters (a=24, b=16, c=37) of this version give slightly
>
> better results in our test than the 2016 version (a=55, b=14, c=36).[2]

These claims about not passing tests can be confirmed by running PractRand on the input, resulting in output like that shown below:

```
RNG_test using PractRand version 0.93
RNG = RNG_stdin64, seed = 0xfac83126
test set = normal, folding = standard (64 bit)

rng=RNG_stdin64, seed=0xfac83126
length= 128 megabytes (2^27 bytes), time= 2.1 seconds
  Test Name                         Raw       Processed     Evaluation
  [Low1/64]BRank(12):256(2)         R= +3748  p~=  3e-1129    FAIL !!!!!!!!
  [Low1/64]BRank(12):384(1)         R= +5405  p~=  3e-1628    FAIL !!!!!!!!
  ...and 146 test result(s) without anomalies
```

Acknowledging the authors go on to say:

> We suggest to use a sign test to extract a random Boolean value[2]

Thus, programmers should prefer the highest bits (e.g., making a heads/tails by writing `random_number < 0` rather than `random_number & 1`). It must be noted, though, that the same test is failed by some instances of the Mersenne Twister and WELL.

The statistical problems extend far beyond the bottom few bits, because it fails the PractRand test even when truncated [3] and fails multiple tests in BigCrush even when the bits are reversed.[4]

Related generators

- xoroshiro128** prevents linear artifacts in the low bits
- xoshiro256+ has 256 bits of state allowing for more parallelism
- xoshiro256** — "our all-purpose, rock-solid generator"

The generators ending with + have weak low bits, so they are recommended for floating point number generation, using only the 53 most significant bits.

See also

- List of Pseudorandom Number Generators
- Pseudorandom Number Generator
- Xorshift
- Mersenne Twister
- WELL

References

1. Blackman, David; Vigna, Sebastiano (2018). "Scrambled Linear Pseudorandom Generators". arXiv:1805.01407 (https://arxiv.org/abs/1805.01407) [cs.DS (https://arxiv.org/archive/cs.DS)].
2. Blackman, David; Vigna, Sebastiano (2018). "Original C source code implementation of xoroshiro128+" (http://prng.di.unimi.it/xoroshiro128plus.c). Retrieved May 4, 2018.
3. "xoroshiro fails PractRand when truncated" (https://www.pcg-random.org/posts/xoroshiro-fails-truncated.html). 2020. Retrieved Dec 30, 2020.
4. "The Xorshift128+ random number generator fails BigCrush" (https://lemire.me/blog/2017/09/08/the-xorshift128-random-number-generator-fails-bigcrush/). 2020. Retrieved Dec 30, 2020.

External links

- Vigna, Sebastiano (2018). "xoshiro / xoroshiro generators and the PRNG shootout" (http://vigna.di.unimi.it/xorshift/). Retrieved 2018-05-04.

- ■