

# 16-bit "798" Xorshift

- original idea: [George Marsaglia](#)
- idea for fast 8-bit implementation: [John Metcalf](#)
- ported by: Veikko Sariola

Xorshift is a fast pseudorandom generator algorithm originally developed by [George Marsaglia](#). [John Metcalf](#) found a 16-bit version of the algorithm that is fast on 8-bit platforms with only single bit shifts available. It has a period of 65535 and passes reasonable tests for randomness. His pseudocode is reprinted here:

```
/* 16-bit xorshift PRNG */

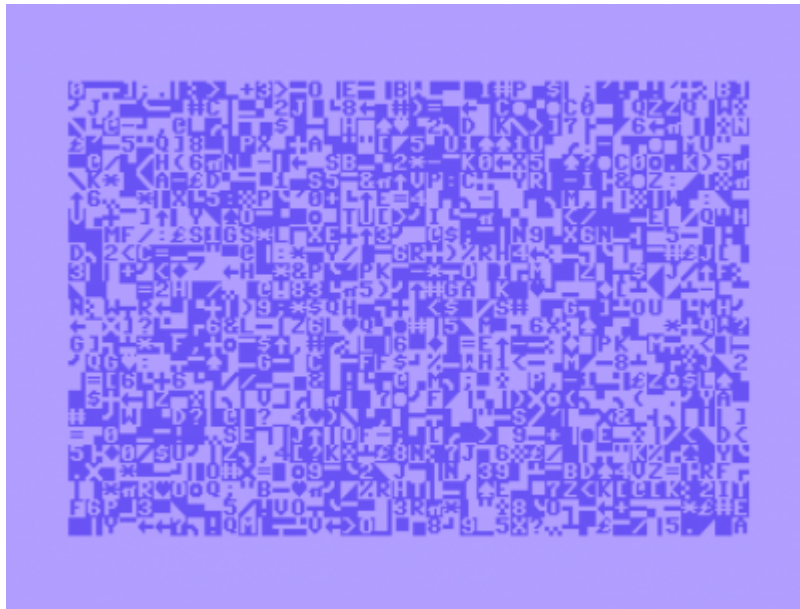
unsigned x = 1;

unsigned xorshift( )
{
    x ^= x << 7;
    x ^= x >> 9;
    x ^= x << 8;
    return x;
}
```

Here is an implementation for the C64. 30 cycles without the RTS.

```
rng_zp_low = $02
rng_zp_high = $03
    ; seeding
    LDA #1 ; seed, can be anything except 0
    STA rng_zp_low
    LDA #0
    STA rng_zp_high
    ...
    ; the RNG. You can get 8-bit random numbers in A or 16-bit numbers
    ; from the zero page addresses. Leaves X/Y unchanged.
random LDA rng_zp_high
    LSR
    LDA rng_zp_low
    ROR
    EOR rng_zp_high
    STA rng_zp_high ; high part of x ^= x << 7 done
    ROR ; A has now x >> 9 and high bit comes from low byte
    EOR rng_zp_low
    STA rng_zp_low ; x ^= x >> 9 and the low part of x ^= x << 7 done
    EOR rng_zp_high
    STA rng_zp_high ; x ^= x << 8 done
    RTS
```

Results:



From:  
<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:  
[https://codebase64.org/doku.php?id=base:16bit\\_xorshift\\_random\\_generator](https://codebase64.org/doku.php?id=base:16bit_xorshift_random_generator)

Last update: **2019-08-12 15:41**

