Philox Engine Correction Proposal

Corrections to `philox_engine` Description for Conformance

Document #:	D0000R1
Date:	2025-04-26
Project:	Programming Language C++
Audience:	Library Evolution Library
Revises:	D0000R0

Reply-to:	Author Juan Lucas Rey <juanlucasrey@gmail.com>

1 Introduction

This document proposes corrections to the description of `philox_engine` as specified in P2075R6, to ensure conformity with expected test outputs and consistency with the original Philox algorithm description.

2 Motivation and Problem Description

The C++ Standard currently specifies:

- For `philox4x32`:
  - Constant template parameters: `0xD2511F53`, `0x9E3779B9`, `0xCD9E8D57`, `0xBB67AE85`
- For `philox4x64`:
  - Constant template parameters: `0xD2E7470EE14C6C93`, `0x9E3779B97F4A7C15`, `0xCA5A826395121157`, `0xBB67AE8584CAA73B`
- For both `philox4x32` and `philox4x64`:
  - Word permutation table: `(0, 3, 2, 1)`
  - Update equations:

$$[\ X_{2k} = \mathrm{mullo}(V_{2k+1}, M_k, w)\ ]$$

$$[\ X_{2k+1} = \mathrm{mulhi}(V_{2k+1}, M_k, w) + ((K_k + q \times C_k) \mathrm{mod} 2^w) + V_{2k}\ ]$$

However, an implementation based on this specification (source) fails the validation test requiring that:

> "If the default-constructed engine is of type `std::philox4x64`, the 10000th consecutive invocation produces the value `3409172418970261260`."

3 Proposed Corrections

3.1 Template Parameters

- For `philox4x32`:
  - Constant template parameters: `0xCD9E8D57`, `0x9E3779B9`, `0xD2511F53`, `0xBB67AE85`
- For `philox4x64`:
  - Constant template parameters: `0xCA5A826395121157`, `0x9E3779B97F4A7C15`, `0xD2E7470EE14C6C93`, `0xBB67AE8584CAA73B`

Note: Only the order of the multiplier constants has changed. This adjustment is consistent with the description in the original Philox paper (Random123, p. 7), where the multipliers for `Philox-4x32` and `Philox-4x64` are listed in a different order.

3.2 Word Permutation Table

Change the permutation table to:

— `(2, 1, 0, 3)`

3.3 Update Equations

Modify the update formulas to:

$$[\ X_{2k} = \text{mulhi}(V_{2k}, M_k, w) + ((K_k + q \times C_k) \bmod 2^w) + V_{2k+1}\ ]$$

$$[\ X_{2k+1} = \text{mullo}(V_{2k}, M_k, w)\ ]$$

An implementation following these modifications (source) passes the specified test and produces the correct 10000th value.

4 Cases for `n = 8` and `n = 16`

The original Philox paper provides no explicit information regarding configurations with `n = 8` or `n = 16`. Therefore, the origin of the permutation tables and constants for these larger values of `n` remains unclear.

We suggest:

— Requesting clarification or provenance for these cases.
— Including specific reference outputs (such as the 10000th invocation result) for `philox8x32`, `philox8x64`, `philox16x32`, and `philox16x64` engines, to ensure future validation.

5 References

— P2075R6 - Add a standard generator based on Philox
— Random123: Parallel Random Number Generation
— Reference implementation and validation tests