Исходными данными для процедуры вычисления хэш-кода $H(M)$ является подлежащее хэшированию сообщение $M \in V^*$ и $IV \in V_{512}$-инициализационный вектор.

Алгоритм вычисления функции H состоит из следующих этапов.

M - произвольный вектор сообщения

IV - начальное значение хэша

Значение инициализационного вектора IV для функции хэширования с длиной хэш-кода 512 бит равно 0^{512}. Значение инициализационного вектора IV для функции хэширования с длиной хэш-кода 256 бит равно $(00000001)^{64}$.

8.1 Этап 1

Присвоить начальные значения текущих величин:

1.1 $h := IV$;

1.2 $N := 0^{512} \in V_{512}$;

1.3 $\Sigma := 0^{512} \in V_{512}$;

1.4 Перейти к этапу 2.

(Предположительно: N - сумма обработанных битов сообщения, Σ - “сумма обработанных блоков”)

Этап 2 - Сокращение хэшируемого сообщения до размера в пределах 512 бит

2.1 Проверить условие $|M| < 512$.

При положительном исходе перейти к этапу 3.

В противном случае выполнить последовательность вычислений по 2.2—2.7.

2.2 Вычислить подвектор $m \in V_{512}$ сообщения M: $M = M'||m$. Далее выполнить последовательность вычислений:

2.3 $h := g_N(h, m)$.

Функция “перемешивания” битов для получения хэша (учитывается уже накопленный хэш и новый блок сообщения)

$$g_N\colon V_{512} \times V_{512} \to V_{512},\ N \in V_{512},$$

значение которой вычисляется по формуле

$$g_N(h, m) = E(LPS(h \oplus N), m) \oplus h \oplus m,$$

где $E(K, m) = X[K_{13}]\, LPSX[K_{12}] \ldots LPSX[K_2]LPSX[K_1](m)$.

Значения $K_i \in V_{512}$, $i = 1,\ldots,13$, вычисляются следующим образом:

$$K_1 = K;$$

$$K_i = LPS(K_{i-1} \oplus C_{i-1}),\ i = 2,\ldots,13.$$

L - линейность (диффузия)
P - перестановка байтов (распространение влияния каждого бита по всему блоку (512 бит))
S - байтовая замена (нелинейность)
E - шифрование с ключом K (13 итераций)

2.4 $N := \mathrm{Vec}_{512}(\mathrm{Int}_{512}(N) \boxplus 512)$.
2.5 $\Sigma := \mathrm{Vec}_{512}(\mathrm{Int}_{512}(\Sigma) \boxplus \mathrm{Int}_{512}(m))$.
2.6 $M := M'$.
2.7 Перейти к шагу 2.1.

Пересчет N, Σ и переход к новому блоку сообщения.

8.3 Этап 3
3.1 $m := 0^{511-|M|}||1||M$.
3.2 $h := g_N(h, m)$.
3.3 $N := \mathrm{Vec}_{512}(\mathrm{Int}_{512}(N) \boxplus |M|)$.
3.4 $\Sigma := \mathrm{Vec}_{512}(\mathrm{Int}_{512}(\Sigma) \boxplus \mathrm{Int}_{512}|m|)$.

Аналогично пункту 2
(приводим сообщение к длине 512
перемешиваем биты
обновляем значения N, Σ)

3.5 $h := g_0(h, N)$.

пересчитываем хэш с учетом всей длины сообщения

$$\text{3.6}\quad h := \begin{cases} g_0(h, \Sigma), & \text{для функции хэширования с длиной хэш-кода 512 бит;} \\ \mathrm{MSB}_{256}(g_0(h, \Sigma)), & \text{для функции хэширования с длиной хэш-кода 256 бит.} \end{cases}$$

пересчитываем хэш с учетом суммы всех блоков, и в случае алгоритма 256 обрезаем до размера 256 (берем первые 256 бит)

Соответствие между реализацией С и Гост
рассмотрим функцию **`streebog_xlps`**
Она соответствует композиции преобразований XLPS

Реализовать функцию, которая составит предвычисленную таблицу для LS-преобразования можно, например, следующим образом:

```
for (j = 0; j < 8; j++)
{
    for (i = 0; i < 256; i++)
    {
        uint8_t t = Pi[i];
        uint64_t a = 0;
        for (k = 0; k < 8; k++)
        {
            if (t & 1)
                a ^= A[63 - 8 * j - k];
            t >>= 1;
        }
        tableLPS[j][i]=a;
    }
}
```

Преобразование осуществляется следующим образом:

$$\mathrm{LPS}\begin{pmatrix}A_7\\A_6\\\vdots\\A_0\end{pmatrix}_{8\times1}=\begin{pmatrix}\mathrm{tableLPS}_{0,a_7}\oplus\mathrm{tableLPS}_{1,a_{15}}\oplus\ldots\oplus\mathrm{tableLPS}_{7,a_{63}}\\\mathrm{tableLPS}_{0,a_6}\oplus\mathrm{tableLPS}_{1,a_{14}}\oplus\ldots\oplus\mathrm{tableLPS}_{7,a_{62}}\\\vdots\\\mathrm{tableLPS}_{0,a_0}\oplus\mathrm{tableLPS}_{1,a_8}\oplus\ldots\oplus\mathrm{tableLPS}_{7,a_{56}}\end{pmatrix}_{8\times1}$$