РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) **RU** (11) **2 726 266**(13) **C1**

(51) МПК
G06F 7/76 (2006.01)
H04L 9/06 (2006.01)

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК
G06F 7/76 (2020.02); H04L 9/06 (2020.02)

(21)(22) Заявка: **2020107680, 20.02.2020**

(24) Дата начала отсчета срока действия патента:
20.02.2020

Дата регистрации:
10.07.2020

Приоритет(ы):
(22) Дата подачи заявки: **20.02.2020**

(45) Опубликовано: **10.07.2020** Бюл. № **19**

Адрес для переписки:
127287, Москва, Старый Петровско-Разумовский пр-д, 1/23, стр. 1, Открытое акционерное общество "Информационные технологии и коммуникационные системы"

(72) Автор(ы):
Рыбкин Андрей Сергеевич (RU)

(73) Патентообладатель(и):
Открытое акционерное общество "Информационные технологии и коммуникационные системы" (RU)

(56) Список документов, цитированных в отчете о поиске: **RU 2598781 C1, 27.09.2016. RU 2219597 C1, 20.12.2003. US 9177666 B2, 03.11.2015. US 7499519 B1, 03.03.2009. US 2004/0150610 A1, 05.08.2004.**

(54) Способ работы регистра сдвига с линейной обратной связью

(57) Реферат:

Изобретение относится к области вычислительной техники. Технический результат заключается в повышении производительности работы РСЛОС типа Фибоначчи при использовании вычислительной системы, позволяющей параллельно вычислять k одинаковых линейных функций от разных аргументов. Технический результат достигается за счет способа работы регистра сдвига с линейной обратной связью (РСЛОС) в вычислительной системе, включающего задание конечного поля P с операцией сложения $\oplus$, операцией умножения $\otimes$, нулевым элементом θ и единичным элементом e; выбор вычислительной системы, имеющей процессор с SIMD-архитектурой, задание натурального числа n; задание натурального числа k, $k \leq n$; задания РСЛОС в конфигурации Фибоначчи, задания количества тактов работы РСЛОС - m, где $m \geq 1$, $m=kv+w$, где v, w - целые неотрицательные числа, $0 \leq w \leq k-1$; осуществление m тактов работы РСЛОС.

RU 2726266 C1

RU 2726266 C1

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 726 266**(13) **C1**

(51) Int. Cl.
G06F 7/76 (2006.01)
H04L 9/06 (2006.01)

(12) ABSTRACT OF INVENTION

(52) CPC
G06F 7/76 (2020.02); H04L 9/06 (2020.02)

(21)(22) Application: **2020107680, 20.02.2020**

(24) Effective date for property rights:
20.02.2020

Registration date:
10.07.2020

Priority:
(22) Date of filing: **20.02.2020**

(45) Date of publication: **10.07.2020** Bull. № **19**

Mail address:
127287, Moskva, Staryj Petrovsko-Razumovskij pr-d, 1/23, str. 1, Otkrytoe aktsionernoe obshchestvo "Informatsionnye tekhnologii i kommunikatsionnye sistemy"

(72) Inventor(s):
Rybkin Andrej Sergeevich (RU)

(73) Proprietor(s):
Otkrytoe aktsionernoe obshchestvo "Informatsionnye tekhnologii i kommunikatsionnye sistemy" (RU)

RU 2726266 C1

(54) OPERATING METHOD OF SHIFT REGISTER WITH LINEAR FEEDBACK

(57) Abstract:

FIELD: computer equipment.

SUBSTANCE: technical result is achieved due to operation method of shift register with linear feedback (LFSR) in computer system, including setting of final field P with operation of addition $\oplus$, a multiplication operation $\otimes$, zero element θ and unit element e; selecting a computing system having a processor with a SIMD architecture, specifying a natural number n; setting a natural number k, $k \leq n$; setting LFSR in a Fibonacci configuration, setting the number of cycles LFSR – m, where $m \geq 1$, $m=kv+w$, where v, w are non-negative integers, $0 \leq w \leq k-1$; m cycles of LFSR operation.

EFFECT: technical result consists in improvement of productivity of LFSR type Fibonacci when using a computer system, which enables parallel calculation of k identical linear functions of different arguments.

1 cl

RU 2726266 C1

Область техники, к которой относится изобретение

Предлагаемое изобретение относится к области вычислительной техники, в к использованию регистров сдвига с линейной обратной связью на вычислительных платформах с SIMD-архитектурой.

Уровень техники

Регистры сдвига с линейной обратной связью (РСЛОС) используются в различных сферах информационных технологий, в том числе в криптографии, сетевых технологиях и цифровой передаче данных. РСЛОС позволяют генерировать линейные рекуррентные последовательности и могут применяться, например, для выработки псевдослучайных последовательностей, вычисления линейных отображений или получения множества различных значений определенной длины.

РСЛОС определяется полем, которому принадлежат элементы входной и выходной последовательности регистра, а также линейной функцией над этим полем, которая задает обратную связь РСЛОС. Различают два типа РСЛОС: РСЛОС в конфигурации Галуа и РСЛОС в конфигурации Фибоначчи. Два этих типа являются эквивалентными, то есть для каждого РСЛОС в конфигурации Галуа может быть построен РСЛОС в конфигурации Фибоначчи (и наоборот), порождающий ту же самую линейную рекуррентную последовательность, возможно, с некоторым сдвигом. В дальнейшем будем рассматривать только РСЛОС в конфигурации Фибоначчи.

Рассмотрим принцип работы РСЛОС, определенного над конечным полем P и имеющего линейную функцию обратной связи РСЛОС $f : P^n \to P$ вида

$f(x_{n-1}, x_{n-2}, \ldots, x_0) = (c_{n-1} \otimes x_{п-1}) \oplus (c_{n-2} \otimes x_{n-2}) \oplus \ldots \oplus (c_0 \otimes x_0)$,

где n - натуральное число,

$c_{i;} \in P$, i=0, 1, …, n-1, - константные элементы поля P,

$x_i \in P$, i=0, 1, …, n-1, - аргументы функции,

«⊗» и «⊕» - операции умножения и сложения в поле P:

• в качестве входной последовательности РСЛОС берут последовательность из n элементов:

$a_0, a_1, \ldots, a_{n-1}, a_i \in P$, i=0, 1, …, n-1;

• формируют начальное состояние РСЛОС, представляющее собой вектор длины n: $(q_{n-1}, q_{n-2}, \ldots, q_0)$,

где $q_i \in P$, i=0, 1, …, n-1,

в виде:

$q_i=a_i$, i=0, 1, …, n-1;

• задают количество тактов работы РСЛОС - m, где m - натуральное число;

• выполняют m тактов работы РСЛОС, причем на s-м такте, $1 \le s \le m$:

○ вычисляют новое состояние РСЛОС, представляющее собой вектор длины n: $(q_{n+s-1}, q_{n+s-2}, \ldots, q_s)$,

где $q_{n+s-1}=f(q_{n+s-2}, q_{n+s-3}, \ldots, q_{s-1}) \in P$,

○ вычисляют элемент выходной последовательности РСЛОС

$b_{s-1} = q_{s-1} \in P$:

• в качестве выходной последовательности РСЛОС берут последовательность из m элементов:

$b_0, b_1, \ldots, b_{m-1}$.

Повсеместное использование РСЛОС делает актуальной задачу получения его

высокопроизводительных реализаций. Одним из базовых способов повышения производительности программных и аппаратных реализаций является использование параллельных вычислений.

Обозначим класс линейных функций, определенных над полем P и имеющих n переменных, через L_n:

$$L_n = \{ l : P^n \to P \mid l(x_{n-1}, x_{n-2}, ..., x_0) =$$

$$= (l_{n-1} \otimes x_{n-1}) \oplus (l_{n-2} \otimes x_{n-2}) \oplus ... \oplus (l_0 \otimes x_0),\ l_i, x_i \in P,\ i = 0, 1, ..., n-1 \}.$$

Рассмотрим вычислительную систему, выполненную с возможностью для любой линейной функции $h \in L_n$ и любых элементов $x_{r,i} \in P$, r=0, 1, …, k-1, i=0, 1, n-1, осуществлять параллельное вычисление k значений $h(x_{0, n-1}, x_{0, n-2}, \ldots, x_{0, 0})$, $h(x_{1, n-1}, x_{1, n-2}, \ldots, x_{1, 0})$, …, $h(x_{k-1, n-1}, x_{k-1, n-2}, \ldots, x_{k-1, 0})$, где k - натуральное число. Обозначим эту вычислительную систему через S_k.

Согласно своему свойству, система S_k позволяет вычислять одновременно k одинаковых линейных функций от разных аргументов. Данное требование является более слабым по сравнению с требованием о возможности параллельного вычисления k произвольных линейных функций от разных аргументов. Это позволяет использовать в качестве вычислительной системы S_k не только системы с возможностью параллельного вычисления k произвольных линейных функций, но и системы, не имеющие данной возможности, но допускающие параллельное вычисление k одинаковых линейных функций.

Примером вычислительной системы S_k может являться вычислительная система с поддержкой SIMD-технологий (Single Instruction, Multiple Data - одна инструкция, несколько блоков данных). В основе SIMD-технологий лежит возможность одновременного выполнения одного и того же преобразования сразу для нескольких фрагментов данных. Таким образом, вычислительная система с поддержкой SIMD-технологий, позволяющая выполнять каждую операцию, применяющуюся при вычислении функции из L_n, одновременно для k фрагментов данных, обеспечивает возможность параллельного вычисления k одинаковых функций из L_n. При этом, в общем случае, такая система не обязана обеспечивать возможность параллельного вычисления k произвольных функций из L_n, поскольку преобразования, используемые при вычислении различных функций, могут отличаться друг от друга, что не позволит или крайне затруднит использование SIMD-технологий для одновременного вычисления этих функций.

Известен способ работы РСЛОС в конфигурации Фибоначчи (патент РФ №2598781, приоритет от 31.07.2015 г.), предусматривающий использование вспомогательных таблиц. Данный способ позволяет на каждом шаге вычислять сразу несколько новых элементов состояния РСЛОС.

Недостатком данного способа является необходимость хранения и использования таблиц, размер которых пропорционален количеству одновременно вычисляемых элементов состояния.

Известен способ работы РСЛОС в конфигурации Фибоначчи в соответствии с традиционным определением РСЛОС, приведенным выше (Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии. 2-е изд., М., Гелиос АРВ, 2002). Способ позволяет на каждом шаге вычислять один новый элемент состояния РСЛОС

посредством вычисления одной линейной функции.

Известный способ принят за прототип.

Недостатком данного способа является неэффективное использование возможностей вычислительной системы S_k, выражающееся в том, что способ подразумевает вычисление только одной линейной функции в произвольный момент времени. В результате, производительность способа на вычислительной системе S_k не зависит от значения k, соответствующего количеству линейных функций, которые могут быть вычислены параллельно на этой системе, и равна производительности способа на вычислительной системе S_1, то есть системе без возможности осуществления параллельных вычислений.

Раскрытие изобретения

Техническим результатом является повышение производительности работы РСЛОС типа Фибоначчи при использовании вычислительной системы, позволяющей параллельно вычислять k одинаковых линейных функций от разных аргументов. При этом предлагаемый способ не требует хранения и использования вспомогательных таблиц.

Предлагаемый способ работы регистра сдвига с линейной обратной связью (РСЛОС) в вычислительной системе, заключается в том, что

- задают конечное поле P с операцией сложения $\oplus$, операцией умножения $\otimes$, нулевым элементом θ и единичным элементом e;
- выбирают вычислительную систему, имеющую процессор с SIMD-архитектурой и выполненную с возможностью
  - преобразования элементов поля P в интерпретируемый вычислительной системой вид и обратного преобразования элементов вида, интерпретируемого вычислительной системой, в элементы поля P;
  - выполнения операций с преобразованными элементами поля P, эквивалентных операциям сложения и умножения в поле P;
- задают натуральное число n;
- задают натуральное число k, $k \le n$;
- задают РСЛОС в конфигурации Фибоначчи, в котором
  - входные и выходные элементы РСЛОС являются элементами поля P;
  - количество элементов вектора состояний РСЛОС равно n;
  - линейная функция обратной связи РСЛОС $f : P^n \to P$ имеет вид

$$f(x_{n-1}, x_{n-2}, \ldots, x_0) = (c_{n-1} \otimes x_{n-1}) \oplus (c_{n-2} \otimes x_{n-2}) \oplus \ldots \oplus (c_0 \otimes x_0),$$

где $c_i \in P$, i=0, 1, …, n-1, - константные элементы поля P,

$x_i \in P$, i=0, 1, …, n-1;

причем

  - при подаче на вход РСЛОС последовательности из n элементов $a_0, a_1, \ldots, a_{n-1}$, где $a_i \in P$, i=0, 1, …, n-1,

начальное состояние РСЛОС, представляющее собой вектор длины n:

$(q_{n-1}, q_{n-2}, \ldots, q_0)$, где $q_i \in P$, i=0, 1, …, n-1,

формируется в виде:

$q_i = a_i$, i=0, 1, …, n-1;

  - в результате выполнения s-го такта работы РСЛОС, $s \ge 1$:
    - новым состоянием РСЛОС становится вектор длины n:

$(q_{n+s-1}, q_{n+s-2}, \ldots, q_s)$,

где $q_{n+s-1}=f(q_{n+s-2}, q_{n+s-3}, \ldots, q_{s-1})\in P$,

- выходным элементом РСЛОС становится элемент

$b_{s-1}=q_{s-1} \in P$;

- задают входную последовательность РСЛОС, состоящую из n элементов поля P: a'_0, a'_1, a'_{n-1}, где $a'_i \in P$, i=0, 1, …, n-1;
- задают количество тактов работы РСЛОС - m, где $m\geq 1$, $m=k\nu+w$, где ν, w - целые неотрицательные числа, $0 \leq w \leq k-1$;
- осуществляют m тактов работы РСЛОС, выполняя следующие действия
  - формируют начальное состояние РСЛОС, представляющее собой вектор длины n:

$(q'_{n-1}, q'_{n-2}, \ldots, q'_s)$, где $q'_i \in P$, i=0, 1, …, n-1,

в виде:

$q'_i=a'_i$, i=0, 1, …, n-1;

  - вычисляют j=0;
  - если ν=0, то переходят к этапу (B);
  - (A) вычисляют с использованием SIMD-инструкций процессора параллельно k элементов $u_{n+jk}, u_{n+jk+1}, \ldots, u_{n+jk+k-1} \in P$:

$u_{n+jk+t}=f(\theta, \theta, \ldots, \theta, q'_{n+jk-1}, q'_{n+jk-2}, \ldots, q'_{jk+t})$, t=0, 1, …, k-1;

  - вычисляют с использованием SIMD-инструкций процессора параллельно k элементов $q'_{n+jk}, q'_{n+jk+1}, \ldots, q'_{n+jk+k-1} \in P$:

$q'_{n+jk+t}=g(\theta, \theta, \ldots, \theta, u'_{n+jk}, u_{n+jk+1}, \ldots, u'_{n+jk+t})$, t=0, 1, …, k-1;

где функция $g : P^k \rightarrow P$ имеет вид

$g(x_{k-1}, x_{k-2}, \ldots, x_0) = (d_{k-1} \otimes x_{k-1}) \oplus (d_{k-2} \otimes x_{k-2}) \oplus \ldots \oplus (d_0 \otimes x_0)$,

где $d_i \in P$, i=0, 1, …, k-1, - константные элементы поля P, для которых справедливо соотношение

$d_i=F_{n-1+i}(e, \theta, \theta, \ldots, \theta)$, i=0, 1, …, k-1,

где функции $F_i : P^n \rightarrow P$, i=0, 1, …, имеют вид

$F_i(x_{n-1}, x_{n-2}, \ldots, x_0)=x_i$, i=0, 1, …, n-1;

$F_i((x_{n-1}, x_{n-2}, \ldots, x_0)=F_{i-1}(f(x_{n-1}, x_{n-2}, \ldots, x_0), x_{n-1}, x_{n-2}, \ldots, x_1)$, i=n, n+1, …;

  - формируют новое состояние РСЛОС, представляющее собой вектор длины n:

$(q'_{n+jk+k-1}, q'_{n+jk+k-2}, \ldots, q'_{n+jk}, q'_{n+jk+-1}, \ldots, q'_{jk+k})$;

  - вычисляют k элементов выходной последовательности РСЛОС

$b'_{jk}, b'_{jk+1}, \ldots, b'_{jk+k-1} \in P$:

$b'_i=q'_i$, i=jk, jk+1, …, jk+k-1;

  - вычисляют j=j+1;
  - если j<ν, то переходят к этапу (A);
  - если w=0, то переходят к этапу (C);
  - (B) вычисляют с использованием SIMD-инструкций процессора параллельно w элементов $u_{n+jk}, u_{n+jk+1}, \ldots, u_{n+jk+w-1} \in P$:

$u_{n+jk+t}=f(\theta, \theta, \ldots, \theta, q'_{n+jk-1}, q'_{n+jk-2}, \ldots, q'_{jk+t})$, t=0, 1, …, w-1;

  - вычисляют с использованием SIMD-инструкций процессора параллельно w элементов $q'_{n+jk}, q'_{n+jk+1}, \ldots, q'_{n+jk+w-1} \in P$:

$q'_{n+jk+t}=g(\theta, \theta, \ldots, \theta, u_{n+jk}, u_{n+jk+1}, \ldots, u_{n+jk+t})$, t=0, 1, …, w-1;

○ формируют новое состояние РСЛОС, представляющее собой вектор длины n:

$(q'_{n+jk+w-1}, q'_{n+jk+w-2}, \ldots, q'_{n+jk}, q'_{n+jk-1}, \ldots, q'_{jk+w})$;

○ вычисляют w элементов выходной последовательности РСЛОС

$b'_{jk}, b'_{jk+1}, \ldots, b'_{jk+w-1} \in P$:

$b'_i=q'_i$, i=jk, jk+1, …, jk+w-1;

• (C) получают выходную последовательность РСЛОС за m тактов работы:

$b'_0, b'_1, \ldots, b'_{m-1} \in P$.

Результат достигается за счет изменения хода вычисления элементов состояния РСЛОС с целью организации возможности их параллельного вычисления.

Рассмотрим вопросы корректности предлагаемого способа. Заметим, что процедура формирования выходных элементов РСЛОС из элементов состояния РСЛОС в предлагаемом способе идентична соответствующей процедуре в способе, выбранном в качестве прототипа. Таким образом, для обоснования корректности предлагаемого способа достаточно показать, что для одной и той же входной последовательности РСЛОС справедливы равенства:

$q'_i=q_i$, i=0, 1, …, n+m-1,

где q'_i - элементы состояния РСЛОС при использовании предлагаемого способа,

q_i - элементы состояния РСЛОС при использовании способа, выбранного в качестве прототипа.

Используемые в предлагаемом способе функции F_i, i=0, 1, …, определяются через композицию функций из L_n и, следовательно, сами принадлежат L_n. Это значит, что для любых $x_j, y_j, c \in P$, j=0, 1, …, n-1, справедливы следующие тождества:

$$F_i(x_{n-1}, x_{n-2}, \ldots, x_0) \oplus F_i(y_{n-1}, y_{n-2}, \ldots, y_0) =$$

$$= F_i(x_{n-1} \oplus y_{n-1}, x_{n-2} \oplus y_{n-2}, \ldots, x_0 \oplus y_0),\ i = 0, 1, \ldots;$$

$$c \otimes F_i(x_{n-1}, x_{n-2}, \ldots, x_0) = F_i(c \otimes x_{n-1}, c \otimes x_{n-2}, \ldots, c \otimes x_0),\ i = 0, 1, \ldots$$

Кроме того, для любых $q_{n+s-1}, q_{n+s-2}, \ldots, q_s \in P$ - элементов состояния РСЛОС при использовании способа, выбранного в качестве прототипа, и любых неотрицательных целочисленных значений s, i, таких что $i+s \le n+m-1$, выполняется:

$$F_i(q_{n+s-1}, q_{n+s-2}, \ldots, q_s) = q_{i+s},\ i \le n-1;$$

$$F_i(q_{n+s-1}, q_{n+s-2}, \ldots, q_s) =$$

$$= F_{i-1}(f(q_{n+s-1}, q_{n+s-2}, \ldots, q_s), q_{n+s-1}, q_{n+s-2}, \ldots, q_{s+1}) =$$

$$= F_{i-1}(q_{n+s}, q_{n+s-1}, q_{n+s-2}, \ldots, q_{s+1}) =$$

$$= F_{i-2}(f(q_{n+s}, q_{n+s-1}, \ldots, q_{s+1}), q_{n+s}, q_{n+s-1}, \ldots, q_{s+2}) =$$

$$= F_{i-2}(q_{n+s+1}, q_{n+s}, q_{n+s-1}, \ldots, q_{s+2}) = \ldots =$$

$$= F_{n-1}(q_{i+s}, q_{i+s-1}, q_{i+s-2}, \ldots, q_{i+s-n+1}) = q_{i+s},\ i \ge n.$$

Заметим, что процедура формирования начального состояния РСЛОС из входных

элементов РСЛОС в предлагаемом способе идентична соответствующей процедуре в способе, выбранном в качестве прототипа. Отсюда следует, что для одной и той же входной последовательности РСЛОС справедливы равенства:

$q'_i=q_i$, i=0, 1, …, n-1.

Примем данный факт за базу индукции. Осуществим шаг индукции, показав, что если для некоторого j, $0 \le j< \nu$, выполняется

$q'_i=q_i$, i=0, 1, …, n+jk-1,

то справедливы равенства:

$q'_i=q_i$, i=n+jk, n+jk+1, n+jk+k-1.

Для этого заметим, что любого t=0, 1, …, k-1 выполняется:

$$q'_{n+jk+t} = g(\theta,\theta,...,\theta,u_{n+jk},u_{n+jk+1},...,u_{n+jk+t}) =$$

$$=(d_0 \otimes u_{n+jk+t}) \oplus (d_1 \otimes u_{n+jk+t-1}) \oplus ... \oplus (d_t \otimes u_{n+jk}) =$$

$$=(F_{n-1}(e,\theta,\theta,...,\theta) \otimes u_{n+jk+t}) \oplus (F_n(e,\theta,\theta,...,\theta) \otimes u_{n+jk+t-1}) \oplus ... \oplus$$

$$\oplus (F_{n+t-1}(e,\theta,\theta,...,\theta) \otimes u_{n+jk}) = F_{n-1}(u_{n+jk+t},\theta,\theta,...,\theta) \oplus$$

$$\oplus F_n(u_{n+jk+t-1},\theta,\theta,...,\theta) \oplus ... \oplus F_{n+t-1}(u_{n+jk},\theta,\theta,...,\theta).$$

Первое слагаемое данной суммы можно представить в виде

$$F_{n-1}(u_{n+jk+t},\theta,\theta,...,\theta) = F_{n-1}(u_{n+jk+t},\theta,\theta,...,\theta) \oplus \theta =$$

$$= F_{n-1}(u_{n+jk+t},\theta,\theta,...,\theta) \oplus F_{n-1}(\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t+1}) =$$

$$= F_{n-1}(u_{n+jk+t},\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t+1}) =$$

$$= F_{n-1}(f(\theta,\theta,...,\theta,q'_{n+jk-1},q'_{n+jk-2},...,q'_{jk+t}),\theta,\theta,...,\theta,$$

$$q_{n+jk-1},q_{n+jk-2},...,q_{jk+t+1}) =$$

$$= F_{n-1}(f(\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t}),\theta,\theta,...,\theta,$$

$$q_{n+jk-1},q_{n+jk-2},...,q_{jk+t+1}) =$$

$$= F_n(\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t}).$$

Прибавляя к полученному значению второе слагаемое, имеем

$$F_n(\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t}) \oplus F_n(u_{n+jk+t-1},\theta,\theta,...,\theta) =$$

$$= F_n(u_{n+jk+t-1},\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t}) =$$

$$= F_n(f(\theta,\theta,...,\theta,q'_{n+jk-1},q'_{n+jk-2},...,q'_{jk+t-1}),\theta,\theta,...,\theta,$$

$$q_{n+jk-1},q_{n+jk-2},...,q_{jk+t}) =$$

$$= F_n(f(\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t-1}),\theta,\theta,...,\theta,$$

$$q_{n+jk-1},q_{n+jk-2},...,q_{jk+t}) =$$

$$= F_{n+1}(\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t-1}).$$

Прибавляя к полученному значению третье слагаемое, имеем

$$F_{n+1}(\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t-1}) \oplus F_{n+1}(u_{n+jk+t-2},\theta,\theta,...,\theta) =$$

$$= F_{n+1}(u_{n+jk+t-2},\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t-1}) =$$

$$= F_{n+1}(f(\theta,\theta,...,\theta,q'_{n+jk-1},q'_{n+jk-2},...,q'_{jk+t-2}),\theta,\theta,...,\theta,$$

$$q_{n+jk-1},q_{n+jk-2},...,q_{jk+t-1}) =$$

$$= F_{n+1}(f(\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t-2}),\theta,\theta,...,\theta,$$

$$q_{n+jk-1},q_{n+jk-2},...,q_{jk+t-1}) =$$

$$= F_{n+2}(\theta,\theta,...,\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+t-2}).$$

Действуя аналогичным образом, на этапе прибавления последнего слагаемого получаем:

$$F_{n+t-1}(\theta,q_{n+jk-1},q_{n+jk-2},...,q_{jk+1}) \oplus F_{n+t-1}(u_{n+jk},\theta,\theta,...,\theta) =$$

$$= F_{n+t-1}(u_{n+jk},q_{n+jk-1},q_{n+jk-2},...,q_{jk+1}) =$$

$$= F_{n+t-1}(f(q'_{n+jk-1},q'_{n+jk-2},...,q'_{jk}),q_{n+jk-1},q_{n+jk-2},...,q_{jk+1}) =$$

$$= F_{n+t-1}(f(q_{n+jk-1},q_{n+jk-2},...,q_{jk}),q_{n+jk-1},q_{n+jk-2},...,q_{jk+1}) =$$

$$= F_{n+t}(q_{n+jk-1},q_{n+jk-2},...,q_{jk}) = q_{n+jk+t}.$$

Таким образом, для любого t=0, 1, …, k-1:

$q'_{n+jk+t}=q_{n+jk+t}$,

то есть

$q'_i=q_i$, i=n+jk, n+jk+1, …, n+jk+k-1.

Последовательно осуществляя приведенный шаг индукции для всех j, $0 \le j \le \nu$, получаем, что

$q'_i=q_i$, i=0, 1, …, n+νk-1.

Обоснование равенства оставшихся w элементов состояния РСЛОС

$q'_i=q_i$, i=n+νk, n+ν+1, …, n+νk+w-1.

выполняется аналогично очередному шагу индукции с тем отличием, что вместо k элементов состояния рассматриваются w элементов состояния.

В результате имеем, что все элементы состояния, получающиеся в процессе работы РСЛОС при использовании предлагаемого способа, равны соответствующим элементам

состояния, получающимся в процессе работы РСЛОС при использовании способа, выбранного в качестве прототипа. С учетом идентичности алгоритмов вычисления выходных значений из элементов состояния в рассматриваемых способах, это доказывает корректность предлагаемого способа работы РСЛОС.

Рассмотрим вопросы эффективности предлагаемого способа.

Выберем вычислительную систему, выполненную с возможностью для любой линейной функции $h \in L_n$ и любых элементов $x_{r,i} \in P$, r=0, 1, k-1,

i=0, 1, n-1, осуществлять параллельное вычисление k значений

$h(x_{0, n-1}, x_{0, n-2}, \ldots, x_{0, 0}), h(x_{1, n-1}, x_{1, n-2}, \ldots, x_{1, 0}), \ldots h(x_{k-1,n-1}, x_{k-1, n-2}, \ldots, x_{k-1, 0})$,

где k - натуральное число.

Возможность параллельного вычисления k функций h подразумевает, что для любого p, $1 \le p \le k$, время параллельного вычисления p функций h равно времени вычисления одной функции h. Обозначим это время через T. Необходимо отметить, что описанная вычислительная система, в том числе, позволяет осуществлять параллельное вычисление k функций $h^* \in L_s$, $1 \le s \le n-1$, ввиду возможности представления функции h^* как функции из L_n путем добавления n-s несущественных переменных.

Оценим время выполнения m последовательных тактов работы РСЛОС, $m \ge 1$, на данной вычислительной системе в случае использования способа, выбранного в качестве прототипа, и в случае использования предлагаемого способа. При использовании способа, выбранного в качестве прототипа, данное время составит

$T_{old}=mT$,

поскольку в этом случае выполнение каждого такта работы РСЛОС потребует одного вычисления функции f.

При использовании предлагаемого способа рассмотрим три возможных варианта значений параметров:

- k=1;
- $k \ge 2$, w=1;
- $k \ge 2$, $w \ne 1$,

где m=kv+w.

При k=1 время выполнения m последовательных тактов работы РСЛОС составит

$$T'_{new} = mT = T_{old},$$

поскольку в этом случае для вычисления каждого нового элемента состояния РСЛОС согласно предлагаемому способу достаточно одного вычисления функции f:

$q'_{n+jk} = g(\theta, \theta, \ldots, \theta, u_{n+jk}) = d_0 \otimes u_{n+jk} = u_{n+jk} = f(q'_{n+jk-1}, q'_{n+jk-2}, \ldots, q'_{jk})$,

так как $d_0=F_{n-1}(e, \theta, \theta, \ldots, \theta)=e$.

При $k \ge 2$, w=1 время выполнения m последовательных тактов работы РСЛОС составит:

$$T''_{new} = v \cdot 2T + T,$$

поскольку в этом случае для вычисления последнего элемента состояния РСЛОС согласно предлагаемому способу достаточно одного вычисления функции f:

$$q'_{n+m-1} = g(\theta, \theta, \ldots, \theta, u_{n+m-1}) = d_0 \otimes u_{n+m-1} =$$

$$= u_{n+m-1} = f(q'_{n+m-2}, q'_{n+m-3}, \ldots, q'_{m-1}),$$

а для вычисления остальных элементов состояния потребуется вычисление функций

f и g на каждые k элементов. В результате, при k ≥ 2, w=1:

$$T_{old} = mT = (kv + w)T \geq (2v + 1)T = v \cdot 2T + T = T''_{new}.$$

При k ≥ 2, w ≠ 1 время выполнения m последовательных тактов работы РСЛОС составит:

$$T_{new} = \lceil m / k \rceil \cdot 2T,$$

поскольку для вычисления каждых k элементов состояния (w элементов состояния на последнем шаге) потребуется вычисление функций f и g. Причем, если w=0, то

$$T_{old} = mT = kvT \geq 2vT = \lceil m / k \rceil \cdot 2T = T_{new}.$$

В случае же w ≥ 2, имеем

$$T_{old} = mT = (kv + w)T \geq (2v + 2)T = (v + 1) \cdot 2T = \lceil m / k \rceil \cdot 2T = T_{new}.$$

Таким образом, производительность предлагаемого способа больше или равна производительности способа, выбранного в качестве прототипа во всех трех рассмотренных случаях. Более того, при k ≥ 2 и достаточно больших значениях m предлагаемый способ позволяет увеличить производительность работы РСЛОС приблизительно в $T_{old} / T_{new} = m / (2\lceil m / k \rceil) \approx k / 2$ раз по сравнению со способом, выбранным в качестве прототипа.

На практике весьма вероятна ситуация, при которой вычисление линейной функции g, зависящей от k аргументов, k ≤ n, может оказаться быстрее вычисления линейной функции f, зависящей от n аргументов. В этом случае разница в производительности способов лишь увеличится.

Обозначим время вычисления функции от k переменных через T_k, а время вычисления функции от n переменных через T_n, $T_k \leq T_n$.

Рассмотрим в данных обозначениях третий вариант значений параметров: k ≥ 2, w ≠ 1, как наиболее общий из всех. В этом случае время выполнения m последовательных тактов работы РСЛОС, m ≥ 1,

• при использовании способа, выбранного в качестве прототипа, составит

$$T^*_{old} = mT_n;$$

• при использовании предлагаемого способа составит:

$$T^*_{new} = \lceil m / k \rceil \cdot (T_n + T_k).$$

Таким образом, при k ≥ 2 и достаточно больших значениях m производительность предлагаемого способа превысит производительность способа, выбранного в качестве прототипа, приблизительно в

$$T^*_{old} / T^*_{new} = mT_n / (\lceil m / k \rceil \cdot (T_n + T_k)) \approx kT_n / (T_n + T_k)$$ раз.

Сделаем еще одно практическое предположение о том, что $T_k/T_n=k/n$. Тогда отношение производительностей способов примет следующий вид:

$$T^*_{old} / T^*_{new} \approx kn / (n + k).$$

Рассмотрим несколько вариантов значений k и соответствующих значений отношения производительности способов:

• при k ≤ n:

$$T^*_{old} / T^*_{new} \approx kn / (n + k) \approx kn / n = k;$$

• при k=n/t, где t - натуральное число:

$$T^*_{old} / T^*_{new} \approx kn / (n+k) = tk^2 / (tk+k) = tk / (t+1) = (t / (t+1)) \cdot k;$$

• при k=n:

$$T^*_{old} / T^*_{new} \approx kn / (n+k) = k^2 / (k+k) = k / 2.$$

В рамках сделанных предположений, производительность предлагаемого способа превысит производительность способа, выбранного в качестве прототипа, в приблизительно от k/2 до k раз. При этом эффективность повышения производительности будет определяться тем, насколько близко значение k к значению n:

$$T^*_{old} / T^*_{new} \approx (t / (t+1)) \cdot k, \text{ где } t = n / k.$$

Осуществление изобретения

Рассмотрим пример реализации предлагаемого способа.

Предлагаемый способ может быть реализован в виде прикладной программы, предназначенной для выполнения на вычислительной системе. В качестве вычислительной системы может использоваться компьютер с процессором, поддерживающим SIMD-вычисления, например, процессор Intel Core i7-2600 с поддержкой SSE-инструкций (Streaming SIMD Extensions) [статья по адресу: https://ark.intel.com/content/www/ru/ru/ark/products/52213/intel-core-i7-2600-processor-8m-cache-up-to-3-80-ghz.html].

Прикладная программа, реализующая работу РСЛОС в конфигурации Фибоначчи согласно предлагаемому способу, может быть составлена специалистом по программированию (программистом).

Рассмотрим поле P с операциями умножения $\otimes$ и сложения $\oplus$, состоящее из 16 элементов и заданное над неприводимым многочленом $X^4 \oplus X \oplus 1, X \in \{0, 1\}$: $P = GF(2^4) = GF(2)[X] / (X^4 \oplus X \oplus 1)$. Для удобства записи будем обозначать элементы поля P целыми числами, предполагая, что элементу поля $(z_3 \otimes X^3) \oplus (z_2 \otimes X^2) \oplus (z_1 \otimes X) \oplus z_0 \in P$, $z_i \in \{0, 1\}$, i=0, 1, 2, 3, соответствует целое число $z_3 \cdot 2^3 + z_2 \cdot 2^2 + z_1 \cdot 2 + z_0 \in Z$.

Для возможности представления элементов поля P в виде, интерпретируемом вычислительной системой, можно использовать взаимно однозначное преобразование элементов поля в двоичные строки, которое сопоставляет элементу поля $(z_3 \otimes X^3) \oplus (z_2 \otimes X^2) \oplus (z_1 \otimes X) \oplus z_0 \in P$, $z_i \in \{0, 1\}$, i=0, 1, 2, 3, двоичную строку $z_3 \| z_2 \| z_1 \| z_0 \in V_4$, где V_s - множество всех двоичных строк длины s, «$\|$» - операция конкатенации двоичных строк.

Для реализации на вычислительной системе операции сложения в поле P можно использовать SSE-инструкцию «pxor», предназначенную для выполнения побитовой операции «исключающее ИЛИ» двух двоичных строк длиной 128. Для реализации на вычислительной системе операции умножения в поле P можно использовать, например, классический алгоритм умножения в столбик без переноса значимого бита с последующим приведением результата умножения по модулю поля $X^4 \oplus X \oplus 1$, или алгоритм, основанный на табличном задании результатов умножения и осуществлении поиска по этим таблицам. В первом случае можно использовать SSE-инструкции «pxor», «pand», предназначенные для выполнения побитовых операций «исключающее ИЛИ»,

«И» двух двоичных строк длиной 128, и SSE-инструкции «psrlw», «psllw», предназначенные для выполнения битовых сдвигов двоичных строк длиной 128 вправо и влево. Во втором случае можно использовать SSE-инструкцию «pshufb», предназначенную для осуществления поиска по заранее вычисленным таблицам.

Зададим значения параметров n=8 и k=4.

Вычисление значения $h(x_7, x_6, \ldots, x_0)$, где $h \in L_8$ - произвольная линейная функция вида:

$h(x_7, x_6, \ldots, x_0) = (h_7 \otimes x_7) \oplus (h_6 \otimes x_6) \oplus \ldots \oplus (h_0 \otimes x_0)$,

$h_i, x_i \in P$, i=0, 1, …, 7,

на вычислительной системе можно осуществлять с использованием упомянутых выше реализаций операций сложения и умножения в поле P посредством SSE-инструкций. Параллельное вычисление четырех значений $h(x_{0,7}, x_{0,6}, \ldots, x_{0,0})$, $h(x_{1,7}, x_{1,6}, \ldots, x_{1,0})$, …, $h(x_{3,7}, x_{3,6}, \ldots, x_{3,0})$, где $x_{r,i} \in P$, r=0, 1, 2, 3, i=0, 1, …, 7, на вычислительной системе можно осуществлять аналогично вычислению одного значения $h(x_7, x_6, \ldots, x_0)$ с предварительной группировкой элементов $x_{0,i}, x_{1,i}, x_{2,i}, x_{3,i}$ на одном 128-битном SSE-регистре с целью одновременного выполнения операций, требуемых при вычислении функции h, сразу для четырех этих элементов.

Рассмотрим РСЛОС в конфигурации Фибоначчи, входные и выходные элементы которого принадлежат конечному полю P, а линейная функция обратной связи $f: P^8 \to P$ задается в виде

$$f(x_7, x_6, \ldots, x_0) = (8 \otimes x_7) \oplus (1 \otimes x_6) \oplus (10 \otimes x_5) \oplus (3 \otimes x_4) \oplus$$

$$\oplus (12 \otimes x_3) \oplus (5 \otimes x_2) \oplus (14 \otimes x_1) \oplus (7 \otimes x_0).$$

Заданные РСЛОС и значение k однозначно определяют функцию $g: P^4 \to P$:

$$g(x_3, x_2, x_1, x_0) = (d_3 \otimes x_3) \oplus (d_2 \otimes x_2) \oplus (d_1 \otimes x_1) \oplus (d_0 \otimes x_0) =$$

$$= (0 \otimes x_3) \oplus (13 \otimes x_2) \oplus (8 \otimes x_1) \oplus (1 \otimes x_0),$$

где элементы $d_i \in P$, i=0, 1, …, 3, вычисляются следующим образом

$d_0 = F_7(1, 0, 0, \ldots, 0) = 1$,

$d_1 = F_8(1, 0, 0, \ldots, 0) = F_7(f(1, 0, 0, \ldots, 0), 1, 0, 0, 0) = f(1, 0, 0, \ldots, 0) = 8 \otimes 1 = 8$,

$d_2 = F_9(1, 0, 0, \ldots, 0) = F_8(f(1, 0, 0, \ldots, 0), 1, 0, 0, \ldots, 0) = F_8(8, 1, 0, 0, \ldots, 0) = F_7(f(8, 1, 0, 0, \ldots, 0), 8, 1, 0, 0, \ldots, 0) = f(8, 1, 0, 0, \ldots, 0) = (8 \otimes 8) \oplus (1 \otimes 1) = 12 \oplus 1 = 13$,

$d_3 = F_{10}(1, 0, 0, \ldots, 0) = F_9(f(1, 0, 0, \ldots, 0), 1, 0, 0, \ldots, 0) = F_9(8, 1, 0, 0, \ldots, 0) = F_8(f(8, 1, 0, 0, \ldots, 0), 8, 1, 0, 0, \ldots, 0) = F_8(13, 8, 1, 0, 0, \ldots, 0) = = F_7(f(13, 8, 1, 0, 0\ldots, 0), 13, 8, 1, 0, 0, \ldots, 0) = f(13, 8, 1, 0, 0, \ldots, 0) = (8 \otimes 13) \oplus (1 \otimes 8) \oplus (10 \otimes 1) = 2 \oplus 8 \oplus 10 = 0$.

Зададим входную последовательность РСЛОС, состоящую из 8 элементов $a'_0, a'_1, \ldots, a'_7$, где $a'_i \in P$, i=0, 1, …, 7:

$a'_0=0$, $a'_1=2$, $a'_2=4$, $a'_3=6$,

$a'_4=9$, $a'_5=11$, $a'_6=13$, $a'_7=15$.

Зададим количество тактов работы РСЛОС m=11. Тогда $m=k\nu+w=4 \cdot 2+3$, то есть $\nu=2$, w=3.

Осуществим 11 тактов работы РСЛОС, для чего выполним следующие действия:

• сформируем начальное состояние РСЛОС, представляющее собой вектор длины

8:

$(q'_7, q'_6, \ldots, q'_0)$, где $q'_i \in P$, i=0, 1, …, 7,

в виде

$(q'_7, q'_6, \ldots, q'_0) = (a'_7, a'_6, \ldots, a'_0) = (15, 13, 11, 9, 6, 4, 2, 0)$;

• вычислим j=0; поскольку v=2 ≠ 0, то

○ вычислим с помощью SIMD-инструкций параллельно 4 элемента $u_8, u_9, u_{11}, \in P$:

$u_8 = f(q'_7, q'_6, \ldots, q'_0) = f(15, 13, 11, 9, 6, 4, 2, 0) = 0,$

$u_9 = f(0, q'_7, q'_6, \ldots, q'_1) = f(0, 15, 13, 11, 9, 6, 4, 2) = 2,$

$u_{10} = f(0, 0, q'_7, q'_6, \ldots, q'_2) = f(0, 0, 15, 13, 11, 9, 6, 4) = 3,$

$u_{11} = f(0, 0, 0, q'_7, q'_6, \ldots, q'_0) = f(0, 0, 0, 15, 13, 11, 9, 6) = 6;$

○ вычислим с помощью SIMD-инструкций параллельно 4 элемента $q'_8, q'_9, q'_{10}, q'_{11}, \in P$

$q'_8 = g(0, 0, 0, u_8) = g(0, 0, 0, 0) = 0,$

$q'_9 = g(0, 0, u_8, u_9) = g(0, 0, 0, 2) = 2,$

$q'_{10} = g(0, u_8, u_9, u_{10}) = g(0, 0, 2, 3) = 0,$

$q'_{11} = g(u_8, u_9, u_{10}, u_{11}) = g(0, 2, 3, 6) = 4;$

○ сформируем новое состояние РСЛОС, представляющее собой вектор длины 8: $(q'_{11}, q'_{10}, \ldots, q'_8, q'_7, q'_4) = (4, 0, 2, 0, 15, 13, 11, 9)$;

○ вычислим 4 выходных элемента РСЛОС $b'_0, b'_1, b'_2, b'_3 \in P$:

$b'_0=q'_0=0, b'_1=q'_1=2, b'_2=q'_2=4, b'_3=q'_3=6;$

• вычислим j=j+1=1; поскольку j=1<2=v, то

○ вычислим с помощью SSE-инструкций параллельно 4 элемента $u_{12}, u_{13}, u_{14}, u_{15} \in P$:

$u_{12} = f(q'_{11}, q'_{10}, \ldots, q'_4) = f(4, 0, 2, 0, 15, 13, 11, 9) = 7,$

$u_{13} = f(0, q'_{11}, q'_{10}, \ldots, q'_5) = f(0, 4, 0, 2, 0, 15, 13, 11) = 10,$

$u_{11} = f(0, 0, q'_{11}, q'_{10}, \ldots, q'_6) = f(0, 0, 4, 0, 2, 0, 15, 13) = 5,$

$u_{15} = f(0 , 0, 0, q'_{11}, q'_{10}, \ldots, q'_7) = f(0, 0, 0, 4, 0, 2, 0, 15) = 13;$

○ вычислим с помощью SSE-инструкций параллельно 4 элемента $q'_{12}, q'_{13}, q'_{14}, q'_{15}, \in P$

$q'_{12} = g(0, 0, 0, u_{12}) = g(0, 0, 0, 7) = 7,$

$q'_{13} = g(0, 0, u_{12}, u_{13}) = g(0, 0, 7, 10) = 7,$

$q'_{14} = g(0, u_{12}, u_{13}, u_{14}) = g(0, 7, 10, 5) = 15,$

$q'_{15} = g(u_{12}, u_{13}, u_{14}, u_{15}) = g(7, 10, 5, 13) = 8;$

○ сформируем новое состояние РСЛОС, представляющее собой вектор длины 8: $(q'_{15}, q'_{14}, \ldots, q'_{12}, q'_{11}, \ldots, q'_8) = (8, 15, 7, 7, 4, 0, 2, 0)$;

○ вычислим 4 выходных элемента РСЛОС $b'_4, b'_5, b'_6, b'_7 \in P$:

$b'_4=q'_4=9, b'_5=q'_5=11, b'_6=q'_6=13, b'_7=q'_7=15;$

• вычислим j=j+1=2; поскольку j=2 ≥ 2=v, то проверим равенство w=0; поскольку w=3 ≠ 0, то

○ вычислим с помощью SSE-инструкций параллельно 3 элемента

$u_{16}, u_{17}, u_{18} \in P$:

$u_{16} = f(q'_{15}, q'_{14}, \ldots, q'_8) = f(8, 15, 7, 7, 4, 0, 2, 0) = 3,$

$u_{17} = f(0, q'_{15}, q'_{14}, \ldots, q'_9) = f(0, 8, 15, 7, 7, 4, 0, 2) = 6,$

$u_{18} = f(0, 0, q'_{15}, q'_{14}, \ldots, q'_{10}) = f(0, 0, 8, 15, 7, 7, 4, 0) = 10;$

○ вычислим с помощью SSE-инструкций параллельно 3 элемента

$q'_{126}\, q'_{17}, q'_{18}, \in P$

$q'_{16} = g(0, 0, 0, u_{16}) = g(0, 0, 0, 3) = 3,$

$q'_{17} = g(0, 0, u_{16}, u_{17}) = g(0, 0, 3, 6) = 13,$

$q'_{18} = g(0, u_{16}, u_{17}, u_{18}) = g(0, 3, 6, 10) = 11;$

○ сформируем новое состояние РСЛОС, представляющее собой вектор длины 8:

$(q'_{18}, q'_{17}, q'_{16}, q'_{15}, \ldots, q'_{11})=(11, 13, 3, 8, 15, 7, 7, 4);$

○ вычислим 3 выходных элемента РСЛОС $b'_8, b'_9, b'_{10} \in P$:

$b'_8=q'_8=0, b'_9=q'_9=2, b'_{10}=q'_{10}=0.$

В результате получим выходную последовательность РСЛОС за 11 тактов работы $b'_0, b'_1, \ldots, b'_{10} \in P$: 0, 2, 4, 6, 9, 11, 13, 15, 0, 2, 0.

Практическое измерение производительности работы рассмотренного РСЛОС, выполненное при больших значениях m ($m \approx 10^7$) в одном потоке одного ядра процессора Intel Core i7-2600, показало, что

• в случае осуществления работы РСЛОС согласно способу, выбранному в качестве прототипа, производительность работы РСЛОС составляет порядка $85 \cdot 10^6$ тактов работы РСЛОС в секунду;

• в случае осуществления работы РСЛОС согласно предлагаемому способу, производительность работы РСЛОС составляет порядка $185 \cdot 10^6$ тактов работы РСЛОС в секунду.

Таким образом, производительностей способов отличается приблизительно в 2,18 раз, что соответствует приведенным теоретическим оценкам, согласно которым предлагаемый способ позволяет увеличить производительность работы РСЛОС в приблизительно от k/2 до k раз.

Рассмотренный РСЛОС и используемые значения параметров выбраны для наглядной демонстрации работы предлагаемого способа. Следует отметить, что предлагаемый способ может быть по аналогии осуществлен и при реализации других РСЛОС в конфигурации Фибоначчи, в том числе РСЛОС, используемых на практике, например, при выработке псевдослучайных последовательностей, вычислении линейных отображений или получении множества различных значений определенной длины.

(57) Формула изобретения

Способ работы регистра сдвига с линейной обратной связью (РСЛОС) в вычислительной системе, заключающийся в том, что

• задают конечное поле P с операцией сложения $\oplus$, операцией умножения $\otimes$, нулевым элементом θ и единичным элементом e;

• выбирают вычислительную систему, имеющую процессор с SIMD-архитектурой и выполненную с возможностью

○ преобразования элементов поля P в интерпретируемый вычислительной системой вид и обратного преобразования элементов вида, интерпретируемого вычислительной

системой, в элементы поля P;

○ выполнения операций с преобразованными элементами поля P, эквивалентных операциям сложения и умножения в поле P;

• задают натуральное число n;

• задают натуральное число k, $k \le n$;

• задают РСЛОС в конфигурации Фибоначчи, в котором

○ входные и выходные элементы РСЛОС являются элементами поля P;

○ количество элементов вектора состояний РСЛОС равно n;

○ линейная функция обратной связи РСЛОС $f : P^n \to P$ имеет вид

$f(x_{n-1}, x_{n-2}, \ldots, x_0) = (c_{n-1} \otimes x_{n-1}) \oplus (c_{n-2} \otimes x_{n-2}) \oplus \ldots \oplus (c_0 \otimes x_0)$,

где $c_i \in P$, i=0, 1, …, n-1, - константные элементы поля P,

$x_i \in P$, i=0, 1 …, n-1;

причем

○ при подаче на вход РСЛОС последовательности из n элементов

$a_0, a_1, \ldots, a_{n-1}$, где $a_i \in P$, i=0, 1, …, n-1,

начальное состояние РСЛОС, представляющее собой вектор длины n:

$(q_{n-1}, q_{n-2}, \ldots, q_0)$, где $q_i \in P$, i=0, 1, …, n-1,

формируется в виде:

$q_i=a_i$, i=0, 1, …, n-1;

○ в результате выполнения s-го такта работы РСЛОС, $s \ge 1$:

▪ новым состоянием РСЛОС становится вектор длины n:

$(q_{n+s-1}, q_{n+s-2}, \ldots, q_s)$,

где $q_{n+s-1}=f(q_{n+s-2}, q_{n+s-3}, \ldots, q_{s-1}) \in P$,

▪ выходным элементом РСЛОС становится элемент

$b_{s-1}=q_{s-1} \in P$;

• задают входную последовательность РСЛОС, состоящую из n элементов поля P:

$a'_0, a'_1, \ldots, a'_{n-1}$, где $a'_i \in P$, i=0, 1, …, n-1;

• задают количество тактов работы РСЛОС - m, где $m \ge 1$, $m=k\nu+w$, где ν, w - целые неотрицательные числа, $0 \le w \le k-1$;

• осуществляют m тактов работы РСЛОС, выполняя следующие действия:

○ формируют начальное состояние РСЛОС, представляющее собой вектор длины n:

$(q'_{n-1}, q'_{n-2}, \ldots, q'_0)$, где $q'_i \in P$, i=0, 1, …, n-1,

в виде:

$q'_i=a'_i$, i=0, 1, …, n-1;

○ вычисляют j=0;

○ если ν=0, то переходят к этапу (B);

○ (A) вычисляют с использованием SIMD-инструкций процессора параллельно k элементов $u_{n+jk}, u_{n+jk+1}, \ldots, u_{n+jk+k-1} \in P$:

$u_{n+jk+t}=f(\theta, \theta, \ldots, \theta, q'_{n+jk-1}, q'_{n+jk-2}, \ldots, q'_{jk+t})$, t=0, 1, …, k-1;

○ вычисляют с использованием SIMD-инструкций процессора параллельно k элементов $q'_{n+jk}, q'_{n+jk+1}, \ldots, q'_{n+jk+k-1} \in P$:

$q'_{n+jk+t}=g(\theta, \theta, \ldots, \theta, u_{n+jk}, u_{n+jk+1}, \ldots, u_{n+jk+t})$, t=0, 1, …, k-1,

где функция $g : P^k \to P$ имеет вид

$g(x_{k-1}, x_{k-2}, \ldots, x_0) = (d_{k-1} \otimes x_{k-1}) \oplus (d_{k-2} \otimes x_{k-2}) \oplus \ldots \oplus (d_0 \otimes x_0)$,

где $d_i \in P$, i=0, 1, …, k-1, - константные элементы поля P, для которых справедливо соотношение

$d_i=F_{n-1+i}(e, \theta, \theta, \ldots, \theta)$, i=0, 1, …, k-1,

где функции $F_i : P^n \to P$, i=0, 1, …, имеют вид

$F_i(x_{n-1}, x_{n-2}, \ldots, x_0)=x_i$, i=0, 1, …, n-1;

$F_i(x_{n-1}, x_{n-2}, \ldots, x_0)=F_{i-1}(f(x_{n-1}, x_{n-2}, \ldots, x_0), x_{n-1}, x_{n-2}, \ldots, x_1)$, i=n, n+1, …;

- формируют новое состояние РСЛОС, представляющее собой вектор длины n: $(q'_{n+jk+k-1}, q'_{n+jk+k-2}, \ldots, q'_{n+jk}, q'_{n+jk-1}, \ldots, q'_{jk+k})$;
- вычисляют k элементов выходной последовательности РСЛОС $b'_{jk}, b'_{jk+1}, \ldots, b'_{jk+k-1} \in P$:

$b'_i=q'_i$, i=jk, jk+1, …, jk+k-1;

- вычисляют j=j+1;
- если j<v, то переходят к этапу (A);
- если w=0, то переходят к этапу (C);
- (B) вычисляют с использованием SIMD-инструкций процессора параллельно w элементов $u_{n+jk}, u_{n+jk+1}, \ldots, u_{n+jk+w-1} \in P$:

$u_{n+jk+t}=f(\theta, \theta, \ldots, \theta, q'_{n+jk-1}, q'_{n+jk-2}, \ldots, q'_{jk+t})$, t=0, 1, …, w-1;

- вычисляют с использованием SIMD-инструкций процессора параллельно w элементов $q'_{n+jk}, q'_{n+jk+1}, \ldots, q'_{n+jk+w-1} \in P$:

$q'_{n+jk+t}=g(\theta, \theta, \ldots, \theta, u_{n+jk}, u_{n+jk+1}, \ldots, u_{n+jk+t})$, t=0, 1, …, w-1;

- формируют новое состояние РСЛОС, представляющее собой вектор длины n: $(q'_{n+jk+w-1}, q'_{n+jk+w-2}, \ldots, q'_{n+jk}, q'_{n+jk-1}, \ldots, q'_{jk+w})$;
- вычисляют w элементов выходной последовательности РСЛОС $b'_{jk}, b'_{jk+1}, \ldots, b'_{jk+w-1} \in P$:

$b'_i=q'_i$, i=jk, jk+1, …, jk+w-1;

- (C) получают выходную последовательность РСЛОС за m тактов работы: $b'_0, b'_1, \ldots, b'_{m-1} \in P$.