

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ DIGITAL PROCESSING OF SIGNALS AND IMAGES

УДК 004.315

Н. И. Червяков, д-р техн. наук, проф., зав. каф., e-mail: k-fmf-primath@stavsu.ru,

П. А. Ляхов, канд. физ.-мат. наук, доц. каф., e-mail: ljahov@mail.ru,

Д. И. Калига, аспирант, ассистент кафедры, e-mail: diana.kalita@mail.ru

Северо-Кавказский федеральный университет, г. Ставрополь

Выбор оптимального набора модулей системы остаточных классов для повышения производительности фильтров с конечной импульсной характеристикой

Представлены результаты исследования производительности различных наборов модулей системы остаточных классов для цифровой обработки сигналов на основе фильтров с конечной импульсной характеристикой (КИХ). Показано, что наилучшую скорость фильтрации обеспечивают модули $\{2^n - 1, 2^n, 2^n + 1, 2^{2^n} - 2^{\frac{n+1}{2}} + 1, 2^{2^n} + 2^{\frac{n+1}{2}} + 1\}$ и $\{2^n - 1, 2^n, 2^n + 1, 2^{2^n + 1} - 1\}$. Разница в производительности между наилучшим и наихудшим наборами модулей достигает 93 %. Полученные результаты могут быть использованы для улучшения характеристик устройств цифровой обработки сигналов в системе остаточных классов и применены для обработки изображений, видео, речи и в других приложениях.

Ключевые слова: система остаточных классов, цифровая обработка сигналов, КИХ-фильтр, набор модулей

Введение

Одним из основных способов цифровой обработки сигналов (ЦОС) является фильтрация. В свою очередь, фильтры подразделяются на нерекурсивные — фильтры с конечной импульсной характеристикой (КИХ-фильтры) и рекурсивные — фильтры с бесконечной импульсной характеристикой (БИХ-фильтры) [1]. КИХ-фильтры используют меньше аппаратных затрат, что делает их проектирование проще по сравнению с БИХ-фильтрами. На сегодняшний день цифровая фильтрация с использованием КИХ-фильтров применяется практически везде, где требуется высокоэффективная обработка сигналов, в частности, в спектральном анализе, обработке изображений, обработке звука, обработке видео и во многих других приложениях [2].

Наиболее перспективным способом повышения производительности систем ЦОС является использование системы остаточных классов (СОК) вместо традиционной двоичной арифметики. Основным преимуществом арифметики остатков по сравнению с позиционной системой счисления (ПСС) является возможность выполнения операций сложения, умножения и вычитания параллельно, благодаря чему повышается скорость работы цифровых устройств [3]. Тем не менее, главным ограничением системы остаточных классов является процесс медленного и сложного выполнения таких опера-

ций, как деление, сравнение, определение знака и обнаружение переполнения, а также восстановление числа по его остаткам [4].

Одним из наиболее важных вопросов теории СОК является определение наилучшего набора модулей для конкретного приложения [5]. Выбор набора модулей играет важную роль в выполнении операций СОК, а также может повлиять на приложения, ее использующие [6]. В данной статье приведены результаты исследования производительности различных модулей СОК для ЦОС на основе КИХ-фильтров.

Введение в СОК

В СОК числа представляются в базе взаимно простых чисел, называемых модулями $\beta = \{p_1, \dots, p_k\}$, $\text{НОД}(p_i, p_j) = 1$, для $i \neq j$. Произведение всех моду-

лей СОК $P = \prod_{i=1}^k p_i$ называется динамическим диа-

пазоном системы. Любое целое число $0 \leq X < P$ может быть единственным образом представлено в СОК в виде вектора $\{x_1, x_2, \dots, x_k\}$, где $x_i = |X|_{p_i} = X \bmod p_i$ [7].

Динамический диапазон СОК обычно делится на две примерно равные части таким образом, чтобы примерно половина диапазона представляла положительные числа, а остальная часть диапазона —

отрицательные. Таким образом, любое целое число, удовлетворяющее одному из двух соотношений

$$-\frac{P-1}{2} \leq X \leq \frac{P-1}{2} \text{ для нечетных } P,$$

$$-\frac{P}{2} \leq X \leq \frac{P}{2} \text{ для четных } P,$$

может быть представлено в СОК.

Операции сложения, вычитания и умножения в СОК определяются формулами

$$A \pm B = (|a_1 \pm b_1|_{p_1}, \dots, |a_k \pm b_k|_{p_k}), \quad (1)$$

$$A \times B = (|a_1 \times b_1|_{p_1}, \dots, |a_k \times b_k|_{p_k}). \quad (2)$$

Равенства (1), (2) показывают параллельную природу СОК, свободную от поразрядных переносов.

Восстановление числа X по остаткам $\{x_1, x_2, \dots, x_k\}$ основано на Китайской теореме об остатках (КТО) [3]

$$X = \left| \sum_{i=0}^k \left| P_i^{-1} \right|_{p_i} x_i P_i \right|_{P}, \quad (3)$$

где $P_i = \frac{P}{p_i}$. Элемент $|P_i^{-1}|_{p_i}$ означает мультипликативный обратный для P_i по модулю p_i .

Существует, однако, и другой способ преобразования числа из СОК в позиционную форму, основанный на применении обобщенной позиционной системы счисления [5]. Число $X < P$ имеет вид $\{x'_1, x'_2, \dots, x'_k\}$, $0 < x'_i \leq p_i$, в обобщенной позиционной системе счисления, если:

$$X = x'_1 + x'_2 p_1 + x'_3 p_1 p_2 + \dots + x'_k \prod_{i=1}^{k-1} p_i, \quad (4)$$

где $x'_i \in [0, p_i)$ — цифры числа X в обобщенной позиционной системе счисления, и

$$x'_1 = x_1 \bmod p_1;$$

$$x'_2 = \{x_2 - x'_1\}_{c_{12}} \bmod p_2;$$

$$x'_3 = ((x_3 - x'_1)_{c_{13}} - x'_2)_{c_{23}} \bmod p_3;$$

$$\dots$$

$$x'_k = (\dots((x_k - x'_1)_{c_{1k}} - x'_2)_{c_{2k}} - \dots$$

$$\dots - x'_{k-1})_{c_{k-i,k}} \bmod p_k.$$

Константы c_{ij} являются мультипликативными обратными элементами для p_i по модулю p_j для всех $1 \leq i \leq j \leq k$, т. е. $c_{ij} p_i \equiv 1 \pmod{p_j}$ для $1 \leq i \leq n$, и могут быть вычислены, например, с помощью алгоритма Евклида.

Таким образом, преимущества представления чисел в СОК могут быть описаны следующим образом [8]:

- так как в СОК отсутствует распространение переноса между арифметическими блоками и числа большой размерности представляются в виде

небольших остатков, это приводит к ускорению в обработке данных;

- при представлении данных с использованием СОК числа большой размерности кодируются в набор небольших остатков, соответственно уменьшается сложность арифметических устройств в каждом канале модуля, что облегчает и упрощает работу вычислительной системы;
- СОК является непозиционной системой без отсутствия зависимости между своими арифметическими блоками, следовательно, ошибка в одном канале не распространяется на другие. Таким образом, облегчается процесс обнаружения и исправления ошибок.

Однако такие операции, как деление, сравнение двух чисел, обнаружение знака являются трудоемкими и дорогими в СОК. Для этих проблемных операций были предложены многие решения. Большинство из них состоит в преобразовании остатка в двоичную систему (обратное преобразование). Вместе с тем, выбор правильного набора модулей является еще одним важным вопросом для построения эффективной СОК с достаточным динамическим диапазоном [9].

Одним из самых известных наборов модулей является набор вида $\{2^n - 1, 2^n, 2^n + 1\}$ [10]. Этот набор известен как средство упрощения расчетов, которые необходимы для осуществления операции обратного преобразования. Тем не менее, арифметические схемы, которые используют модуль $(2^n + 1)$ имеют большую задержку среди всех трех каналов. Таким образом, для того чтобы упростить сложность, вызванную данным модулем, были предложены новые наборы модулей: $\{2^{n-1} - 1, 2^n - 1, 2^n\}$ [11] и $\{2^n - 1, 2^n, 2^n + 1 - 1\}$ [12]. Эти три набора имеют $3n$ -битный динамический диапазон. Так как для многих приложений ЦОС необходим больший динамический диапазон, были предложены новые наборы модулей $\{2^n - 1, 2^n, 2^{2n+1} - 1\}$ [13] и $\{2^n - 1, 2^n + 1, 2^{2n} + 1\}$ [14], которые обеспечивают $4n$ -битный диапазон, а также $\{2^n - 1, 2^{2n}, 2^{2n} + 1\}$ [15], который дает $5n$ -битный динамический диапазон. Несмотря на то что динамический диапазон стал больше, увеличилась задержка работы арифметических устройств, основанных на этих наборах, в связи с использованием модулей с большими величинами. Чтобы устранить этот недостаток и сохранить большой динамический диапазон, были предложены наборы, имеющие от четырех до пяти модулей, например, $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ [16], $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$ [16], $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ [17], $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ [18], $\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\}$ [19],

$$\{2^n - 1, 2^n + 1, 2^{2n} - 1, 2^{2n+1} - 3\} [20],$$

$$\{2^n - 1, 2^{2n}, 2^n + 1, 2^{2n} + 1\} [18],$$

$$\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1, 2^{n+1} + 1\} [21]$$

и $\{2^n, 2^{\frac{n}{2}} - 1, 2^{\frac{n}{2}} + 1, 2^n + 1, 2^{2n-1} - 1\} [22].$

Каждый из этих наборов имеет свои преимущества и недостатки. Некоторые из них предлагают большой динамический диапазон, в то время как другие имеют большую степень параллелизма. Некоторые из них могут привести к более эффективному обратному преобразованию, в то время как другие — к получению более эффективного остатка арифметических устройств [23]. Как будет видно далее, выбор наиболее эффективного набора модулей играет существенную роль в скорости работы КИХ-фильтра.

Проектирование КИХ-фильтров в СОК

Любой КИХ-фильтр можно задать следующим выражением:

$$y_n = \sum_{k=0}^N b_k x_{n-k} \quad (6)$$

где x_n — входная последовательность сигнала; b_k — коэффициенты фильтра; N — порядок фильтра; y_n — последовательность сигнала, полученного на выходе фильтра [24].

Из формулы (6) видно, что для работы КИХ-фильтра используются только операции сложения и умножения, что позволяет при его реализации использовать положительные качества СОК.

Рассмотрим низкочастотный КИХ-фильтр 15-го порядка, АЧХ которого представлена на рис. 1. Проектирование фильтра было осуществлено в MATLAB® с использованием встроенного приложения Filter Design and Analysis. Фильтр, АЧХ которого изображена на рис. 1, построен с использованием алгоритма Паркса—МакКлеллана [25]. Коэффициенты фильтра в формате двойной точности и в целочисленном формате данных приведены в табл. 1.

Для получения целочисленных значений данного фильтра была использована команда

```
num2int(quantizer([12, 11]), z)
```

Таким образом, полученные целочисленные значения коэффициентов фильтра являются 12-битными числами, из которых 11 бит — величина числа, 1 бит — знак. Эти целочисленные значения в дальнейшем могут быть представлены в СОК путем простого вычисления остатков от деления на модули СОК.

Так как исходные коэффициенты фильтра являются числами двойной точности длиной 64 бит, то переход от такого представления к 12-битному влечет за собой ошибку округления. На рис. 2 показана АЧХ этой ошибки. Из рисунка видно, что максимум ошибки примерно на 25 дБ ниже, чем область записания фильтра.

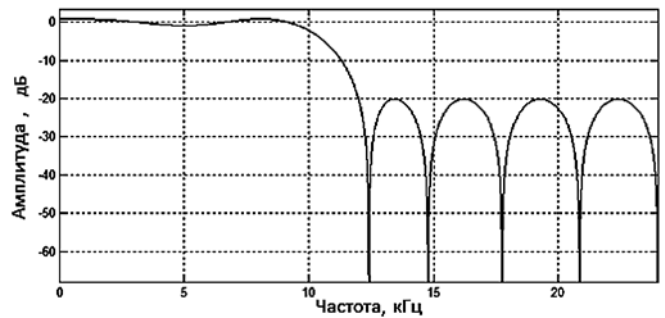


Рис. 1. АЧХ низкочастотного фильтра 15-го порядка

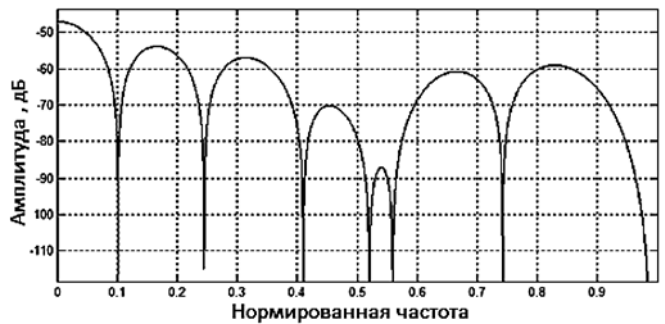


Рис. 2. АЧХ ошибки округления КИХ-фильтра

Таблица 1

Коэффициенты КИХ-фильтра 15-го порядка

Коэффициенты фильтра	Числовой формат представления	
	Двойная точность (64 бит)	Целое число (12 бит)
$b_0 = b_{15}$	-0,04842854521480	-100
$b_1 = b_{14}$	0,031641772569045	64
$b_2 = b_{13}$	0,066305238639482	135
$b_3 = b_{12}$	0,016109250214128	32
$b_4 = b_{11}$	-0,076170627278596	-156
$b_5 = b_{10}$	-0,041780626517982	-86
$b_6 = b_9$	0,183762552952241	376
$b_7 = b_8$	0,417189375297071	854

Для равномерного распределения последствий из-за возникновения ошибок округления между фильтром и сигналом рассмотрим случай, когда в качестве входной последовательности используется сигнал с такой же точностью, что и коэффициенты фильтра, т. е. 12 бит. Тогда диапазон работы КИХ-фильтра равен:

$$\max\{|y(n)|\} = 2^{12} \sum_{k=0}^{15} |b_k| = 4096 \cdot 3606 = 14\,770\,176. \quad (7)$$

Передаточная функция фильтра, описывающая связь между сигналами на входе и выходе фильтра, задается в виде

$$H(z) = b_0 + b_1 z^{-1} + \dots + b_{15} z^{-15} = \sum_{i=0}^{15} b_i z^{-i}. \quad (8)$$

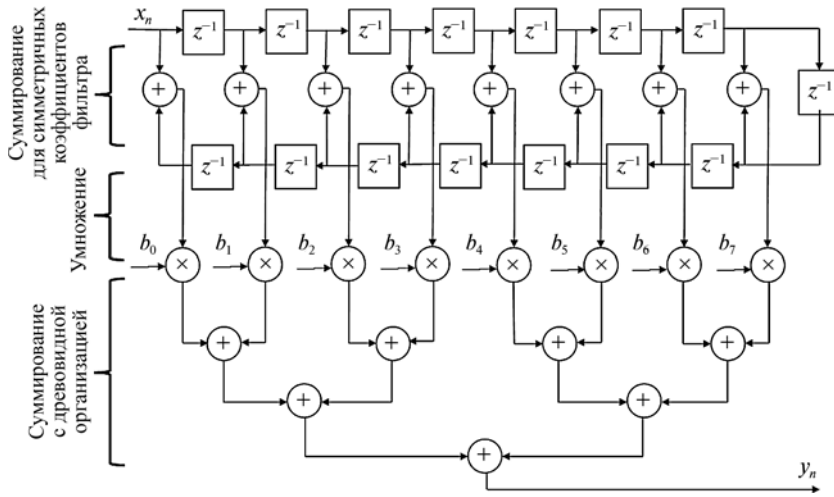


Рис. 3. Схема работы фильтра 15-го порядка

С учетом того, что коэффициенты построенного фильтра симметричны, выражение (8) можно переписать в следующем виде:

$$H(z) = \sum_{i=0}^8 b_k(z^k + z^{15-k}). \quad (9)$$

На рис. 3 представлена схема фильтра, описанного формулой (9), где блок z^{-1} — это блок задержки. При реализации изображенной схемы фильтра в СОК достаточно заменить коэффициенты b_p , $i = 0, \dots, 15$, на их модулярное представление, а блоки сложения и умножения — на блоки сложения по модулю и умножения по модулю соответственно.

Исследование наборов модулей СОК для фильтрации

Для оценки времени работы КИХ-фильтров в СОК проводился подсчет тактов синхронизации вычислительной системы, необходимых для выполнения полного цикла работы фильтра при обработке одного вновь поступившего отсчета входной последовательности. В КИХ-фильтрах в СОК используются только операции сложения и умножения, которые в свою очередь тем или иным способом (в зависимости от конкретного вида модулей СОК) сводится к комбинации базовых логических элементов. В соответствии с методом, предложенным в работе [26], использовались следующие значения числа тактов синхронизации вычислительной системы для базовых элементов:

- время работы двухвходовых элементов "И", "ИЛИ", "И-НЕ", "ИЛИ-НЕ" принимается равным $T_1 = 1$;
- время работы двухвходовых элементов "ИСКЛЮЧАЮЩЕЕ ИЛИ", а также мультиплексора 2:1 принимается равным $T_2 = 2$;
- время работы двоичного полусумматора принимается равным $T_3 = 2$;

- время работы полного двоичного сумматора принимается равным $T_4 = 4$. Как уже отмечалось ранее, выбор набора модулей играет важную роль в улучшении производительности КИХ-фильтров, построенных в СОК. Проведем анализ производительности КИХ-фильтра с использованием наиболее известных из них. В табл. 2 приведено время задержки при модулярном сложении и умножении для каждого рассмотренного набора модулей.

Для каждого из приведенных в табл. 2 наборов модулей было подобрано наименьшее n , обеспечивающее необходимый диапазон, указанный в формуле (7), т. е. все наборы модулей должны быть взаимно простыми и удовлетворять условию $P \geq 14\,770\,176$. В табл. 3 приведены минимальные значения n , удовлетворяющие этому условию.

В наборе модулей $\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1, 2^{n-1} + 1\}$ (см. табл. 2) при наименьшем необходимом $n = 5$ не выполняется требование взаимной простоты. Поэтому этот набор был исключен из исследования. При реализации фильтра 15-го порядка логическая глубина выполнения операций равна 5, при этом используется одно параллельное умножение и 4 параллельных сложения (см. рис. 3). Задержка для модулярного сложения (рис. 4) и умножения получена с учетом данных из табл. 2. В по-

Таблица 2

Время выполнения операций модулярного сложения и умножения для разных наборов модулей СОК

Набор модулей	Задержка, такты	
	Модулярное сложение	Модулярное умножение
$\{2^n - 1, 2^n, 2^n + 1\}$	$8n + 11$	$16n + 12$
$\{2^{n-1} - 1, 2^n - 1, 2^n\}$	$8n$	$16n - 7$
$\{2^n - 1, 2^n, 2^{n+1} - 1\}$	$8n + 8$	$16n + 9$
$\{2^n - 1, 2^n, 2^{2n+1} - 1\}$	$16n + 8$	$32n + 9$
$\{2^n - 1, 2^n + 1, 2^{2n} + 1\}$	$16n + 11$	$32n + 12$
$\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$	$8n + 11$	$16n + 12$
$\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$	$8n + 19$	$16n + 28$
$\{2^n - 1, 2^{2n} - 1, 2^{2n} + 1\}$	$16n + 11$	$32n + 12$
$\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$	$16n + 11$	$32n + 12$
$\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$	$16n + 8$	$32n + 9$
$\{2^n - 1, 2^n, 2^n + 1, 2^{2n} - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\}$	$8n + 11$	$24n + 13$
$\{2^n - 1, 2^n, 2^n + 1, 2^{n-1} - 1, 2^{n+1} + 1\}$	$8n + 19$	$16n + 28$
$\{2^n, 2^{\frac{n}{2}} - 1, 2^{\frac{n}{2}} + 1, 2^n + 1, 2^{n-1} - 1\}$	$16n - 8$	$32n - 23$
$\{2^n - 1, 2^n + 1, 2^{2n} - 1, 2^{2n+1} - 3\}$	$16n + 11$	$48n + 13$
$\{2^n - 1, 2^{2n}, 2^n + 1, 2^{2n} + 1\}$	$16n + 11$	$28n + 12$

Время (в тактах) выполнения модульных операций для фильтрации в СОК с различными наборами модулей

№ набора модулей	Набор модулей	n	Десятичная запись модулей	Задержка, такты		Итого
				+	×	
1	$\{2^n - 1, 2^n, 2^n + 1\}$	8	{255,256,257}	75	140	440
2	$\{2^{n-1} - 1, 2^n - 1, 2^n\}$	9	{255,511,512}	72	137	425
3	$\{2^n - 1, 2^n, 2^{n+1} - 1\}$	7	{127,128,255}	64	121	377
4	$\{2^n - 1, 2^n, 2^{2n+1} - 1\}$	6	{63,64,8191}	104	201	617
5	$\{2^n - 1, 2^n + 1, 2^{2n+1}\}$	6	{63,65,4097}	107	204	632
6	$\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$	6	{63,64,65,127}	59	108	344
7	$\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$	7	{127,128,129,257}	75	140	440
8	$\{2^n - 1, 2^{2n} - 1, 2^{2n} + 1\}$	5	{32,1023,1025}	91	187	551
9	$\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$	5	{31,32,33,1025}	91	287	651
10	$\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$	5	{31,32,33,2047}	88	169	521
11	$\{2^n - 1, 2^n, 2^n + 1, 2^{2n} - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\}$	5	{31,32,33,25,41}	51	133	337
12	$\{2^n, 2^{\frac{n}{2}} - 1, 2^{\frac{n}{2}} + 1, 2^{2n} + 1, 2^{n-1} - 1\}$	6	{64,7,9,65,2047}	88	169	521
13	$\{2^n - 1, 2^n + 1, 2^{2n} - 1, 2^{2n+1} - 3\}$	4	{15,17,254,509}	75	205	505
14	$\{2^n - 1, 2^{2n}, 2^n + 1, 2^{2n} + 1\}$	4	{15,256,17,257}	75	124	424

следней колонке табл. 3 приведен суммарный результат числа необходимых тактов на выполнение всех модульных операций при фильтрации в СОК. Анализ полученных результатов позволяет сделать вывод, что наиболее эффективными наборами модулей для ЦОС являются $\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\}$ и $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$. Наименее эффективными наборами модулей, т. е. такими, которые будут приносить наибольшую задержку при выполнении модульных операций, являются наборы $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ и $\{2^n - 1, 2^n + 1, 2^{2n} + 1\}$.

Среди всех трехмодульных наборов наилучший результат дает набор $\{2^n - 1, 2^n, 2^{n+1} - 1\}$, его задержка составляет 377 тактов, наихудшим является набор $\{2^n - 1, 2^n + 1, 2^{2n} + 1\}$ с задержкой в 632 такта. Наиболее эффективным среди четырехмодульных наборов является набор $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ с задержкой, равной 344 такта, а наименее эффективным — набор $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ с задержкой в 651 такт. Для пятимодульных наборов наименьшую задержку в 337 тактов дает набор $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\}$, наибольшую в 521 такт — набор $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$.

Примем время работы самого быстрого набора модулей $\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\}$, равным 100 %. В табл. 4 показано относительное время работы разных наборов модулей СОК, в сравнении с этим набором модулей. Самый эффектив-

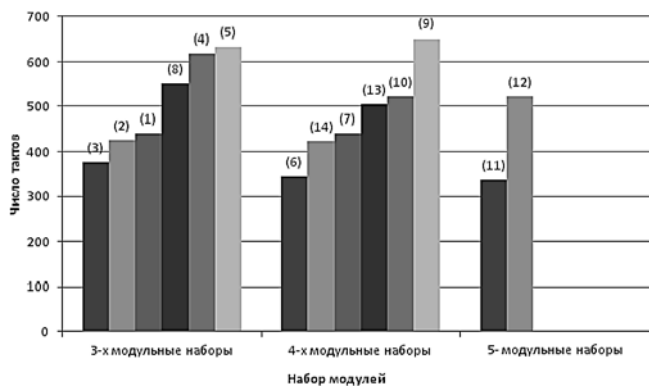


Рис. 4. Модульная задержка наборов модулей. Набор модулей соответствует порядковому номеру в табл. 3

Таблица 4

Сравнение времени фильтрации в СОК для разных наборов модулей

Набор модулей	Итоговая задержка, %
$\{2^n - 1, 2^n, 2^n + 1, 2^{2n} - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\}$	100
$\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$	102
$\{2^n - 1, 2^n, 2^{n+1} - 1\}$	112
$\{2^{n-1} - 1, 2^n - 1, 2^n\}$	126
$\{2^n - 1, 2^{2n}, 2^n + 1, 2^{2n} + 1\}$	126
$\{2^n - 1, 2^n, 2^n + 1\}$	131
$\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$	131
$\{2^n - 1, 2^n + 1, 2^{2n} - 1, 2^{2n+1} - 3\}$	150
$\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$	155
$\{2^n, 2^{\frac{n}{2}} - 1, 2^{\frac{n}{2}} + 1, 2^{2n} + 1, 2^{n-1} - 1\}$	155
$\{2^n - 1, 2^{2n} - 1, 2^{2n} + 1\}$	163
$\{2^n - 1, 2^n, 2^{2n+1} - 1\}$	183
$\{2^n - 1, 2^n + 1, 2^{2n} + 1\}$	188
$\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$	193

ный набор модулей $\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\}$ на 93 % работает быстрее, чем самый медленный набор $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$, что подтверждает важность задачи выбора наиболее эффективного набора модулей СОК для ЦОС.

Заключение

В статье проведен анализ скорости работы КИХ-фильтров при использовании различных наборов модулей СОК. Наиболее эффективными оказались

наборы $\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\}$ и $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$. Наихудшими наборами являются наборы $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ и $\{2^n - 1, 2^n + 1, 2^{2n} + 1\}$, работающие на 93 % и 88 % соответственно медленнее, чем набор модулей $\{2^n - 1, 2^n, 2^n + 1, 2^n - 2^{\frac{n+1}{2}} + 1, 2^n + 2^{\frac{n+1}{2}} + 1\}$.

Дальнейшая работа по исследованию эффективности различных наборов модулей СОК при реализации цифровых фильтров может быть связана с непосредственной реализацией на программируемых логических интегральных схемах и интегральных схемах специального назначения и оценкой параметров скорости работы, энергопотребления и аппаратных затрат.

Интересным направлением дальнейших исследований в области применения КИХ-фильтров в СОК является поиск наиболее эффективных наборов модулей, обеспечивающих высокую скорость выполнения модульных операций для конкретных приложений. Применение оптимального набора модулей в СОК для обработки изображений, видео, речи, спутниковой связи и других приложениях позволит существенно ускорить обработку данных.

Работа выполнена при финансовой поддержке РФФИ, грант № 14-07-31004-мол-а.

Авторы выражают благодарность рецензенту за ценные замечания к рукописи статьи.

Список литературы

1. **Tan L., Jiang J.** Digital Signal Processing, Second Edition: Fundamentals and Applications // Academic Press. 2013. 876 p.
2. **Shahana T. K., James R. K., Jose B. R., Jacob K. P., Sasi S.** Performance Analysis of FIR Digital Filter Design: RNS Versus Traditional // ISCIT 2007 International Symposium on Communications and Information Technologies Proceedings. 2007. P. 1–5.
3. **Omondi A., Premkumar B.** Residue Number Systems: Theory and Implementation // Imperial College Press. 2007. P. 296.
4. **Cardarilli G. C., Nannarelli A., Re M.** Residue number system for low-power DSP applications // Proc. 41st Asilomar Conf. Signals, Syst., Comput. 2007. P. 1412–1416.
5. **Червяков Н. И., Сахнюк П. А., Шапошников А. В., Макоха А. Н.** Нейрокомпьютеры в остаточных классах. М.: Радиотехника, 2003. 272 с.

6. **Stamenković N.** Digital FIR Filter Architecture Based on the Residue Number System // Facta Universitatis, Ser.: Elec. Energ. 2009. Vol. 22, N. 1. P. 125–140.
7. **Afsheh A., Mojoodi A.** An improved reverse converter for moduli set $(2^n - 1, 2^n, 2^n + 1)$ // ISCIT 2010 International Symposium on Communications and Information Technologies Proceedings. 2010. P. 928–933.
8. **Червяков Н. И., Ляхов П. А.** Реализация КИХ-фильтров в системе остаточных классов // Нейрокомпьютеры: разработка, применение. 2012. № 5. С. 15–24.
9. **Goh V. T., Siddiqi M. U.** Multiple Error Detection and Correction Based on Redundant Residue Number Systems // IEEE Transactions on Communications. 2008. Vol. 56, N. 3. P. 325–330.
10. **Piestrak S. J.** A high-speed realization of a residue to binary number system converter // IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing. 1995. Vol. 42. P. 661–663.
11. **Wang W., Swamy M. N., Ahmad M. O., Wang Y.** A high-speed residue-to-binary converter for three-moduli $(2^k, 2^k - 1, 2^{k-1} - 1)$ RNS and a scheme for its VLSI implementation // IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing. 2000. Vol. 47. P. 1576–1581.
12. **Mohan P. V. A.** RNS-to-binary converter for a new three moduli set $\{2^{n+1} - 1, 2^n, 2^n - 1\}$ // IEEE Trans. on Circuits and Systems-II: Express Briefs. 2007. Vol. 54. P. 775–779.
13. **Molahosseini A. S., Navi K., Rafsanjani M. K.** A new residue to binary converter based on mixed-radix conversion // 3rd International Conference on ICTTA. 2008. P. 1–6.
14. **Wang W., Swamy M. N. S., Ahmad O., Wang Y.** A study of the residue-to-binary converters for the three-moduli sets // IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications. 2003. Vol. 50. P. 235–243.
15. **Hariri A., Navi K., Rastegar R.** A new high dynamic range moduli set with efficient reverse converter // Computers & Mathematics with Applications Journal. 2008. Vol. 55. P. 660–668.
16. **Mohan P. V. A., Premkumar A. B.** RNS-to-binary converters for two four-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} - 1\}$ and $\{2^n - 1, 2^n, 2^n + 1, 2^{n+1} + 1\}$ // IEEE Trans. on Circuits and Systems-I: Regular Papers. 2007. Vol. 54. P. 1245–1254.
17. **Cao B., Chang C. H., Srikanthan T.** An efficient reverse converter for the 4-moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{2n} + 1\}$ // IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications. 2003. Vol. 50. P. 1296–1303.
18. **Molahosseini A. S., Navi K., Dadkhah C., Kavehei O., Timarichi S.** Efficient reverse converter designs for the new 4-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{2n+1} - 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2n}, 2^{2n} + 1\}$ based on new CRTs // IEEE Trans. on Circuits and Systems-I: Regular Papers. 2010. Vol. 57. P. 823–835.
19. **Hiasat A. A.** VLSI implementation of new arithmetic residue to binary decoders // IEEE Trans. on VLSI Systems. 2005. Vol. 13. P. 153–158.
20. **Zhang W., Siy P.** An efficient design of residue to binary converter for four moduli set $\{2^n - 1, 2^n + 1, 2^{2n} - 2, 2^{2n+1} - 3\}$ based on new CRT-II // Information Sciences Journal. 2008. Vol. 178. P. 264–279.
21. **Cao B., Chang C. H., Srikanthan T.** A residue-to-binary converter for a new five-moduli set // IEEE Trans. on Circuits and Systems-I: Regular Papers. 2007. Vol. 54. P. 1041–1049.
22. **Molahosseini A. S., Dadkhah C., Navi K.** A new five moduli set for efficient hardware implementation of the reverse converter // IEICE Electronics Express. 2009. Vol. 6. P. 1006–1012.
23. **Antro S. F., Bajard J. C., Sousa L.** RNS based Elliptic Curve Point Multiplication for Massive Parallel Architectures // Computer Journal. Oxford University Press. 2011.
24. **Стемпковский А. Л., Корнилов А. И., Семенов М. Ю.** Особенности реализации устройств цифровой обработки сигналов в интегральном исполнении с применением модулярной арифметики // Информационные технологии. 2004. № 2. С. 2–9.
25. **Ingle V. K., Proakis J. K.** Digital Signal Processing Using MATLAB® // ser. BookWare Companion Series, New York, Madrid: Brooks/Cole Publishing Company. 1999. 218 p.
26. **Younes D., Steffan P. A.** Comparative Study on Different Moduli Sets in Residue Number System // Dept. of Microelectronics Brno University of Technology Brno, Czech Republic, 2013.

N. I. Chervyakov, Professor, Head of Department, e-mail: k-fmf-primath@stavs.ru,
P. A. Lyakhov, Associate Professor, e-mail: ljahov@mail.ru,
D. I. Kalita, Graduate Student, Assistant of the Department, e-mail: diana.kalita@mail.ru
North-Caucasus Federal University, Stavropol

Selection of the Optimal RNS Moduli to Improve FIR Filters Performance

The paper presents the results of a study the performance of different Residue Number System moduli sets for digital signal processing based on the Finite Impulse Response filters. It is shown that the best filtration rate provide by moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{2^n} - 2^{\frac{n+1}{2}} + 1, 2^{2^n} + 2^{\frac{n+1}{2}} + 1\}$ and $(2^n - 1, 2^n, 2^n + 1, 2^{2^n+1} - 1)$. The difference in performance between the best and the worst moduli sets up to 93 %. The results can be used to improve the performance of the digital signal processing devices used Residue Number System and applied to the processing of images, video, speech and other applications.

Keywords: residue number system, digital signal processing, FIR filter, moduli set

References

1. **Tan L., Jiang J.** *Digital Signal Processing, Second Edition: Fundamentals and Applications*, Academic Press, 2013. 876 p.
2. **Shahana T. K., James R. K., Jose B. R., Jacob K. P., Sasi S.** Performance Analysis of FIR Digital Filter Design: RNS Versus Traditional, *ISCIT 2007 International Symposium on Communications and Information Technologies Proceedings*, 2007, pp. 1–5.
3. **Omondi A., Premkumar B.** Residue Number Systems: Theory and Implementation, Imperial College Press, 2007. 296 p.
4. **Cardarilli G. C., Nannarelli A., Re M.** Residue number system for low-power DSP, applications, *Proc. 41st Asilomar Conf. Signals, Syst., Comput*, 2007, pp. 1412–1416.
5. **Chervyakov N. I., Sakhyuk P. A., Shaposhnikov A. V., Makokha A. N.** *Neirokomp'yutery v ostatochnykh klassakh* [Neurocomputers in residue numbers], Moscow: Radiotekhnika, 2003. 272 p. (In Russian).
6. **Stamenković N.** Digital FIR Filter Architecture Based on the Residue Number System, *Facta Universitatis, Ser.: Elec. Energ*, 2009, vol. 22, no. 1, pp. 125–140.
7. **Afsheh A., Mojoodi A.** An improved reverse converter for moduli set $\{2^n - 1, 2^n, 2^n + 1\}$, *ISCIT 2010 International Symposium on Communications and Information Technologies Proceedings*, 2010, pp. 928–933.
8. **Chervyakov N. I., Lyakhov P. A.** Realizatsiya KIX-fil'trov v sisteme ostatochnykh klassov [Implementation of FIR filters in Residue Number System], *Neirokomp'yutery: razrabotka, primeneniye*, 2012, no. 5, pp. 15–24. (In Russian).
9. **Goh V. T., Siddiqi M. U.** Multiple Error Detection and Correction Based on Redundant Residue Number Systems, *IEEE Transactions on Communications*, 2008, vol. 56, no. 3, pp. 325–330.
10. **Piestrak S. J.** A high-speed realization of a residue to binary number system, converter, *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, 1995, vol. 42, pp. 661–663.
11. **Wang W., Swamy M. N., Ahmad M. O., Wang Y.** A high-speed residue-to-binary converter for three-moduli $(2^k, 2^k - 1, 2^{k-1} - 1)$ RNS and a scheme for its VLSI implementation, *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, 2000, vol. 47, pp. 1576–1581.
12. **Mohan P. V. A.** RNS-to-binary converter for a new three moduli set $\{2^{n+1} - 1, 2^n, 2^n - 1\}$, *IEEE Trans. on Circuits and Systems-II: Express Briefs*, 2007, vol. 54, pp. 775–779.
13. **Molahosseini A. S., Navi K., Rafsanjani M. K.** A new residue to binary converter based on mixed-radix conversion, *3rd International Conference on ICTTA*, 2008, pp. 1–6.
14. **Wang W., Swamy M. N. S., Ahmad O., Wang Y.** A study of the residue-to-binary converters for the three-moduli sets, *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications*, 2003, vol. 50, pp. 235–243.
15. **Hariri A., Navi K., Rastegar R.** A new high dynamic range moduli set with efficient reverse converter. *Computers & Mathematics with Applications Journal*, 2008, vol. 55, pp. 660–668.
16. **Mohan P. V. A., Premkumar A. B.** RNS-to-binary converters for two four-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{2^n+1} - 1\}$ and $\{2^n - 1, 2^n, 2^n + 1, 2^{2^n+1} + 1\}$, *IEEE Trans. on Circuits and Systems-I: Regular Papers*, 2007, vol. 54, pp. 1245–1254.
17. **Cao B., Chang C. H., Srikanthan T.** An efficient reverse converter for the 4-moduli set $\{2^n - 1, 2^n, 2^n + 1, 2^{2^n} + 1\}$, *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications*, 2003, vol. 50, pp. 1296–1303.
18. **Molahosseini A. S., Navi K., Dadkhah C., Kavehei P., Timarchi S.** Efficient reverse converter designs for the new 4-moduli sets $\{2^n - 1, 2^n, 2^n + 1, 2^{2^n+1} - 1\}$ and $\{2^n - 1, 2^n + 1, 2^{2^n}, 2^{2^n} + 1\}$ based on new CRTs, *IEEE Trans. on Circuits and Systems-I: Regular Papers*, 2010, vol. 57, pp. 823–835.
19. **Hiasat A. A.** VLSI implementation of new arithmetic residue to binary decoders, *IEEE Trans. on VLSI Systems*, 2005, vol. 13, pp. 153–158.
20. **Zhang W., Siy P.** An efficient design of residue to binary converter for four moduli set $\{2^n - 1, 2^n + 1, 2^{2^n} - 2, 2^{2^n+1} - 3\}$ based on new CRT-II, *Information Sciences Journal*, 2008, vol. 178, pp. 264–279.
21. **Cao B., Chang C. H., Srikanthan T.** A residue-to-binary converter for a new five-moduli set, *IEEE Trans. on Circuits and Systems-I: Regular Papers*, 2007, vol. 54, pp. 1041–1049.
22. **Molahosseini A. S., Dadkhah C., Navi K.** A new five moduli set for efficient hardware implementation of the reverse converter, *IEICE Electronics Express*, 2009, vol. 6, pp. 1006–1012.
23. **Antro S. F., Bajard J. C., Sousa L.** RNS based Elliptic Curve Point Multiplication for Massive Parallel Architectures, *Computer Journal*, Oxford University Press, 2011.
24. **Stempkovskii A. L., Kornilov A. I., Semenov M. Yu.** Osobennosti realizatsii ustroystv tsifrovoy obrabotki signalov v integral'nom ispolnenii s primeneniem modulyarnoi arifmetiki [Features of digital signal processing devices realization integrally with the use of modular arithmetic], *Informatsionnye tekhnologii*, 2004, no. 2, pp. 2–9. (In Russian).
25. **Ingle V. K., Proakis J. K.** *Digital Signal Processing Using MATLAB®*, ser. BookWare Companion Series, New York, Madrid: Brooks/Cole Publishing Company, 1999. 218 p.
26. **Younes D., Steffan P.** A Comparative Study on Different Moduli Sets in Residue Number System, *Dept. of Microelectronics Brno University of Technology Brno*, Czech Republic, 2013.