

УДК 004.383.3

doi: 10.21685/2072-3059-2023-1-3

**Использование системы остаточных классов с модулями  
вида  $\{2^n - 1, 2^n, 2^n + 1\}$  для снижения аппаратных  
затрат цифрового фильтра**

**М. В. Бергерман**

Северо-Кавказский центр математических исследований, Ставрополь, Россия

maxx07051997@inbox.ru

**Аннотация.** *Актуальность и цели.* Низкая производительность устройств цифровой фильтрации является главной проблемой для решения практических задач цифровой обработки сигналов. Одним из способов повышения производительности таких устройств является параллельная обработка сигналов с использованием арифметики системы остаточных классов. *Материалы и методы.* Предложена архитектура усеченного умножителя с накопителем и преобразователей для вычисления по модулю  $(2^n + 1)$ . Также был предложен способ фильтрации сигналов цифрового фильтра с конечной импульсной характеристикой с использованием усеченных умножителей с накопителем в системе остаточных классов с модулями специального вида  $\{2^n - 1, 2^n, 2^n + 1\}$ . *Результаты.* Использование модулей  $\{2^n - 1, 2^n, 2^n + 1\}$  позволяет снизить аппаратные затраты по сравнению с модулями  $\{2^n - 1, 2^n\}$ . Результаты аппаратного моделирования показали, что для 32-разрядного фильтра в системе остаточных классов предложенный набор модулей позволяет снизить аппаратные затраты до 32,5 %, но при этом увеличивается задержка до 16,4 %. *Выводы.* Полученные результаты открывают возможность для создания устройств с низкими аппаратными затратами.

**Ключевые слова:** цифровая обработка сигналов, цифровой фильтр, система остаточных классов, умножители с накопителем, аппаратное моделирование

**Благодарности:** исследование проведено в Северо-Кавказском центре математических исследований в рамках соглашения с Министерством науки и высшего образования Российской Федерации (соглашение № 075-02-2023-938).

**Для цитирования:** Бергерман М. В. Использование системы остаточных классов с модулями вида  $\{2^n - 1, 2^n, 2^n + 1\}$  для снижения аппаратных затрат цифрового фильтра // Известия высших учебных заведений. Поволжский регион. Технические науки. 2023. № 1. С. 32–43. doi: 10.21685/2072-3059-2023-1-3

**Using the residue number system  
with moduli of  $\{2^n - 1, 2^n, 2^n + 1\}$  type to reduce  
the hardware costs of the digital filter**

**M.V. Bergerman**

North-Caucasus Center for Mathematical Research, Stavropol, Russia

maxx07051997@inbox.ru

**Abstract.** *Background.* The low performance of digital filtering devices is the main problem for solving practical problems of digital signal processing. One of the ways to improve the performance of such devices is parallel signal processing using residue number system arithmetic. *Materials and methods.* The article proposes architectures of a truncated multiply and accumulate unit and converters for calculating modulo  $2^n + 1$ . A method for filtering digital filter signals with a finite impulse response using truncated multiply and accumulate unit in residue number system with moduli of a special type  $\{2^n - 1, 2^n, 2^n + 1\}$  was also proposed. *Results.* The use of moduli  $\{2^n - 1, 2^n, 2^n + 1\}$  makes it possible to reduce hardware costs compared to moduli  $\{2^n - 1, 2^n\}$ . The results of hardware simulation showed that for a 32-bit filter in residue number system, the proposed moduli set allows reducing hardware costs to 32.5%, but at the same time, the delay increased to 16.4%. *Conclusions.* The results obtained open up the possibility of creating devices with low hardware costs.

**Keywords:** digital signal processing, digital filter, residue number system, multiply and accumulate unit, hardware simulation

**Acknowledgement:** The research was carried out at the North Caucasus Center for Mathematical Research under an agreement with the Ministry of Science and Higher Education of the Russian Federation (agreement No. 075-02-2023-938).

**For citation:** Bergerman M.V. Using the residue number system with moduli of  $\{2^n - 1, 2^n, 2^n + 1\}$  type to reduce the hardware costs of the digital filter. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki = University proceedings. Volga region. Engineering sciences.* 2023;(1):32–43. (In Russ.). doi: 10.21685/2072-3059-2023-1-3

## Введение

На сегодня цифровая обработка сигналов (ЦОС) применяется на ключевых отраслях современной промышленности, таких как цифровое телевидение, средства массовой информации, медицина и многие другие [1]. Основным средством цифрового преобразования сигналов являются цифровые фильтры (ЦФ), которые разделяются на фильтры с конечной импульсной характеристикой (КИХ) и бесконечной импульсной характеристикой (БИХ). На практике часто используются КИХ-фильтры за счет устойчивости и отсутствия обратной связи, что позволяет избежать ошибки при вычислениях [2]. КИХ-фильтры используются для решения задач вычислений с целью обучения нейронных сетей [3], для задач очистки изображений от шума [4], для задач шифрования данных [5] и многих других.

Повышение производительности ЦФ является главной задачей, стоящей перед учеными на сегодня. Одним из способов увеличения производительности цифровых устройств являются параллельные вычисления. Система остаточных классов (СОК) является такой системой счисления, где вычисления производятся параллельно по нескольким каналам (модулям) СОК [6]. Достоинствами СОК являются параллельные вычисления: операции сложения, вычитания и умножения в СОК выполняются по каждому основанию независимо друг от друга, а также снижение разрядности по каждому вычислительному каналу, что позволяет снизить аппаратные и временные затраты. Однако существуют недостатки, связанные с использованием так называемых немодульных операций в СОК, таких как деление, определение знака, сравнение и т.д. [7].

### 1. Цифровая фильтрация с конечной импульсной характеристикой

Результатом фильтрации в цифровом фильтре с конечной импульсной характеристикой является сигнал, который вычисляется по формуле

$$T(N) = \sum_{i=0}^K c_i S(N-i), \quad (1)$$

где  $c_i$  – коэффициент фильтра;  $S$  – исходный сигнал;  $S(N)$  – отсчет исходного сигнала;  $K$  – порядок фильтра.

Для реализации КИХ-фильтров используются блоки умножителей с накопителем (МАС), которые выполняют операцию умножения с последующей операцией сложения с предыдущим результатом вычисления:

$$T_i = c_i S(N-i) + T_{i-1}. \quad (2)$$

Большинство задач с использованием фильтрации решаются устройствами, работающими в режиме реального времени, для которых скорость вычислений, аппаратные ресурсы и затрачиваемая энергия играют важную роль. Поэтому возникает потребность в улучшении эксплуатационных показателей устройств цифровой фильтрации.

### 2. Система остаточных классов

Система остаточных классов – это непозиционная система счисления, состоящая из набора чисел, называемых основаниями или модулями системы  $\{d_1, d_2, \dots, d_s\}$ . Все модули  $d_i$  должны быть попарно взаимно просты, т.е. должно выполняться следующее условие:  $\text{НОД}(d_j, d_l) = 1, j \neq l$ . Произведение модулей образует динамический диапазон системы  $D$ , который определяет число возможных комбинаций остатков. Прямое преобразование из позиционной системы счисления (ПСС) в СОК осуществляется путем взятия остатка по модулю:

$$X = (x_1, x_2, \dots, x_s) = (X \bmod d_1, X \bmod d_2, \dots, X \bmod d_s), \quad (3)$$

$$Y = (y_1, y_2, \dots, y_s) = (Y \bmod d_1, Y \bmod d_2, \dots, Y \bmod d_s). \quad (4)$$

Расчеты в СОК выполняются следующим образом:

$$X * Y = (x_1 * y_1, x_2 * y_2, \dots, x_s * y_s), \quad (5)$$

где  $*$  – модульная операция: сложение, умножение или вычитание.

Обратное преобразование из СОК в ПСС основано на Китайской теореме об остатках:

$$Z = \left| \sum_{i=1}^s \left| D_i^{-1} \right|_{d_i} z_i \right|_{D_i} \Big|_D, \quad (6)$$

где  $D_i = \frac{D}{d_i}$ ;  $\left| D_i^{-1} \right|_{d_i}$  является мультипликативной инверсией относительно  $D_i$ .

Главный вопрос состоит в том, какой набор модулей для фильтрации в СОК будет самым эффективным в плане аппаратных и временных затрат. Одним из примеров является использование модулей вида  $2^n$ ,  $2^n - 1$ . Их эффективность достигается за счет алгоритмов, позволяющих получать результаты с низкой задержкой и занимать меньше площади по сравнению с модулями общего вида [8]. Однако данные модули ограничиваются числом возможных комбинаций наборов модулей, так как необходимо соблюдать попарно взаимную простоту: модуль  $2^n$  можно использовать лишь раз, а модуль  $(2^n - 1)$  чаще всего ограничивается числами Мерсенна, где  $n$  – простое число. Для вычислений в СОК совместно с модулями  $2^n$ ,  $2^n - 1$  используют также модуль вида  $(2^n + 1)$ . Главным недостатком модуля вида  $(2^n + 1)$  являются большие затраты на выполнение арифметических операций, так как для вычислений необходимо использовать  $(n+1)$ -бит. Для вычислений с  $n$ -битной разрядностью по модулю  $(2^n + 1)$  используется специальная техника вычитания единицы. Данная техника позволяет выполнять вычисления по данному модулю более эффективно [9].

### 3. Техника вычитания единицы

Техника вычитания единицы преобразовывает остаток из  $(n+1)$ -битного значения в  $n$ -битное и позволяет выполнять вычисления в  $n$ -битной разрядности. Преобразование заключается в следующем: из значения остатка, кроме «нуля», вычитается единица, а для нуля значение меняется только на старшем  $n$ -бите – определителе нуля, т.е. выполняется условие

$$x' = \begin{cases} x - 1, & \text{if } x \neq 0; \\ 100\dots 00, & \text{if } x = 0. \end{cases} \quad (7)$$

Сложение и умножение с данной техникой выполняются по формулам:

$$x' + y' + \overline{C_{out}} = s', \quad (8)$$

$$x' \times y' + x' + y' = p', \quad (9)$$

где  $x'$  и  $y'$  являются слагаемыми или множителями;  $s'$  и  $p'$  являются результатами сложения и умножения соответственно, которые представлены в технике вычитания единицы;  $\overline{C_{out}}$  – инверсия старшего бита переноса.

Для обратного преобразования в СОК в исходное значение по модулю  $(2^n + 1)$  достаточно сложить значение  $x'$  от 0 до  $(n-1)$  бита с инверсией знака определителя нуля  $\overline{x'_n}$ , расположенный на  $n$ -бите числа  $x'$ :

$$x_{n:0} = x'_{n-1:0} + \overline{x'_n}, \quad (10)$$

где  $x_{n:0}$  – исходное значение остатка по модулю  $(2^n + 1)$ , которое имеет  $(n+1)$ -битную разрядность  $(n:0)$ ,  $x'_{n-1:0}$  – число в технике вычитания единицы с разрядностью в  $n$  бит  $(n-1:0)$ .

#### 4. Архитектуры предложенных преобразователей, усеченного множителя с накопителем и сумматора по модулю $(2^n + 1)$

В качестве базовых блоков сложения используются полные сумматоры (full adder, FA) и полусумматоры (half adder, HA) [10].

Чтобы выполнять вычисления по модулю  $(2^n + 1)$ , для начала необходимо преобразовать остатки с использованием техники вычитания единицы. Для выполнения вычитания чисел необходимо вычитаемое число представить в дополнительном коде [10], а затем сложить его с исходным остатком по модулю  $(2^n + 1)$ . Значение числа «-1» в дополнительном коде равно " $\underbrace{11\dots 11}_{n\text{-бит}}$ ". Если исходное значение равно нулю, то сумма будет равна " $11\dots 11$ ", а необходимое значение должно быть равно " $\underbrace{10\dots 00}_{(n+1)\text{-бит}}$ ". Отсюда

следует, что необходимо отслеживать все биты, чтобы узнать, нулевое это значение или нет.

В качестве сумматора используется архитектура сумматора с последовательным переносом (carry propagate adder, CPA) (рис. 1). Так как второе слагаемое нам известно, схема упрощается за счет свойств логических выражений:

$$S = A \oplus 1 \oplus D = \bar{A} \oplus D = A \odot D; \quad (11)$$

$$C = ((A \oplus 1) \wedge D) \vee (A \wedge 1) = (\bar{A} \wedge D) \vee A = A \vee D, \quad (12)$$

где  $A$ ,  $B$  и  $D$  – слагаемые;  $S$  и  $C$  – результаты бит суммы и бит переноса;  $\oplus$ ,  $\odot$ ,  $\wedge$ ,  $\vee$  и  $\bar{A}$  – логические элементы «Исключающее ИЛИ» (Exclusive-OR, XOR), «Исключающее ИЛИ-НЕ» (Exclusive NOR, XNOR), «И» (AND), «ИЛИ» (OR) и «НЕ» (NOT). Архитектура сумматора для вычисления  $S$  и  $C$  представлена на рис. 2. Бит переноса на последнем блоке сумматора будет показывать, какое число было исходным, и в зависимости от данного бита выполняется преобразование остальных бит. Если старший бит переноса будет равен '1', то исходное число не было равно "0" и старший бит результата будет равен '0'. Если старший бит переноса станет равным '0', значит, исходное число было нулевым и старший бит результата будет равен '1', а остальные биты обнулятся. Для коррекции преобразования используется логическая операция «И» на каждом бите, кроме знака, так как по ее свойству  $A \wedge 1 = A$ ;  $A \wedge 0 = 0$ . Старший бит вычисляется с использованием логической операции «НЕ» для старшего бита переноса. На рис. 3 представлена архитектура преобразователя вычитания единицы.

Для перевода значения из техники вычитания единицы в исходное значение СОК по модулю  $(2^n + 1)$  используется архитектура последовательных НА, так как достаточно сложить число в технике вычитания единицы без старшего бита определителя нуля с инверсией этого же бита. Схема обратного преобразователя вычитания единицы показана на рис. 4.

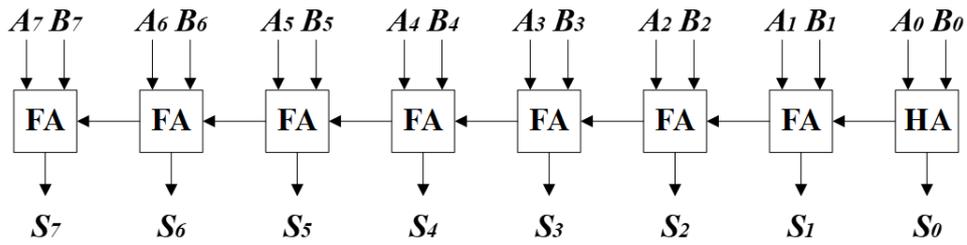


Рис. 1. Архитектура сумматора CPA по модулю  $2^8$

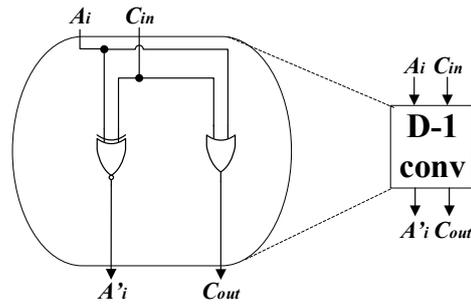


Рис. 2. Схема вычисления для перевода числа в технике вычитания единицы

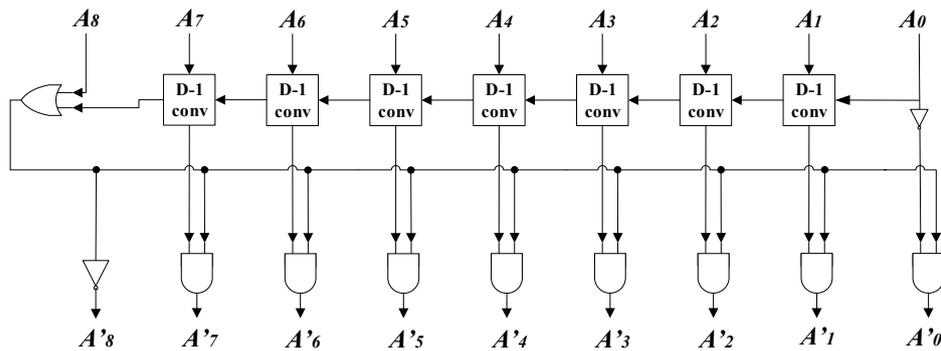


Рис. 3. Схема перевода числа в технике вычитания единицы, основанный на CPA

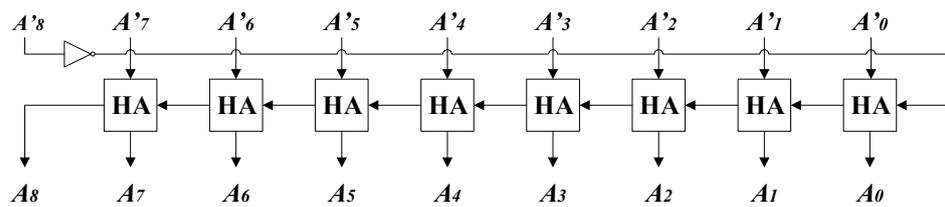


Рис. 4. Схема перевода числа из техники вычитания единицы 8-битного числа, основанной на HA

Для фильтрации в СОК по модулям  $2^n$  и  $(2^n - 1)$  используются блоки ТМАС и ТМАС с техникой обратного переноса (end-around carry ТМАС,

ЕАС-ТМАС) [8] соответственно, но вместо сумматора Когге-Стоуна будет использоваться сумматор CPA по модулю  $2^n$ , состоящий из блоков НА и FA (рис. 1) для модуля  $2^n$  и сумматор с последовательным переносом с техникой обратного переноса (end-around carry CPA, ЕАС-CPA) (рис. 5) для модуля  $(2^n - 1)$ , так как данные сумматоры расходуют меньше аппаратных затрат.

Для модуля  $(2^n + 1)$  будут использоваться преобразователи техники вычитания единицы, блоки ТМАС с техникой инверсного обратного переноса (Inverted ЕАС-ТМАС, ІЕАС-ТМАС), состоящие из сумматоров, с сохранением переноса с техникой инверсного обратного переноса (ІЕАС-carry save adder, ІЕАС-СSA) и результирующий сумматор с последовательным переносом с техникой инверсного обратного переноса (ІЕАС-CPA) [11].

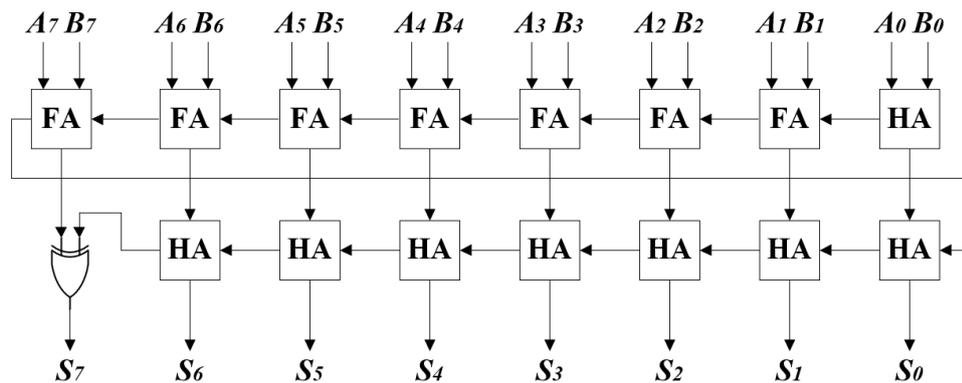


Рис. 5. Архитектура сумматора ЕАС-CPA по модулю  $(2^8 - 1)$

Как и у блоков ТМАС и ЕАС-ТМАС, в ІЕАС-ТМАС также имеются генератор частичных произведений (рис. 6) и дерево сумматоров CSA (CSA-tree) (рис. 7). Но в ІЕАС-ТМАС выполняется больше операций сложения за счет большего числа слагаемых, в том числе и преобразования вычитания единицы, а также используются мультиплексоры для проверки знаков двух множителей. Если знак одного из множителей равен "1" (0 в технике вычитания единицы), то результатом будут два слагаемых, которые обращаются в нуль, если это первый блок ТМАС, или слагаемые с предыдущего ІЕАС-ТМАС (рис. 8) в остальных случаях. В данных вычислениях не отслеживается старший бит, который определяет нулевое значение. Он будет вычисляться на последнем этапе суммирования оставшихся двух чисел.

Последнее суммирование двух чисел выполняется за счет предложенного сумматора ІЕАС-CPA. В отличие от сумматора ЕАС-CPA, здесь старший бит переноса инвертируется и подается на последовательные блоки НА. В первом блоке НА и в каждом блоке FA необходимо отслеживать «сумму» двух чисел  $(A \oplus B)$ . Если на каждом бите сумма будет равна "1", то общая сумма двух чисел образует значение "0" в преобразовании вычитания единицы и старший бит будет равен "1", в остальных случаях бит будет равен "0". Архитектуры блоков FA и НА с отслеживанием суммы представлены на рис. 9. Архитектура сумматора ІЕАС-CPA показана на рис. 10.

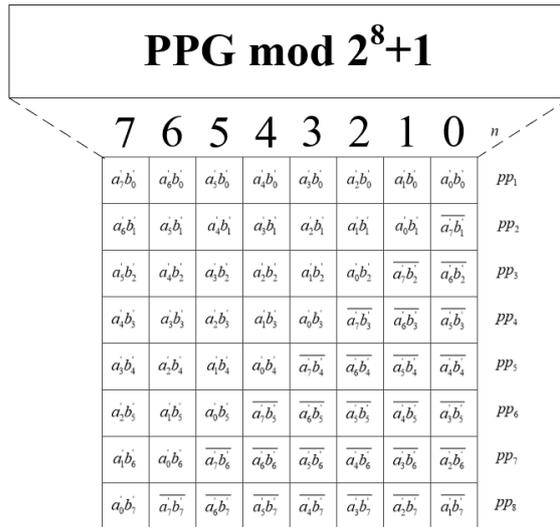


Рис. 6. Генерация частичных произведений по модулю  $(2^8 + 1)$

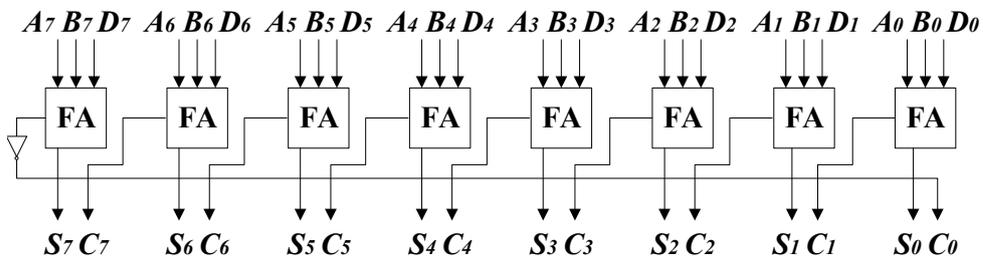


Рис. 7. Архитектура 8-битного сумматора IEAC-CSA

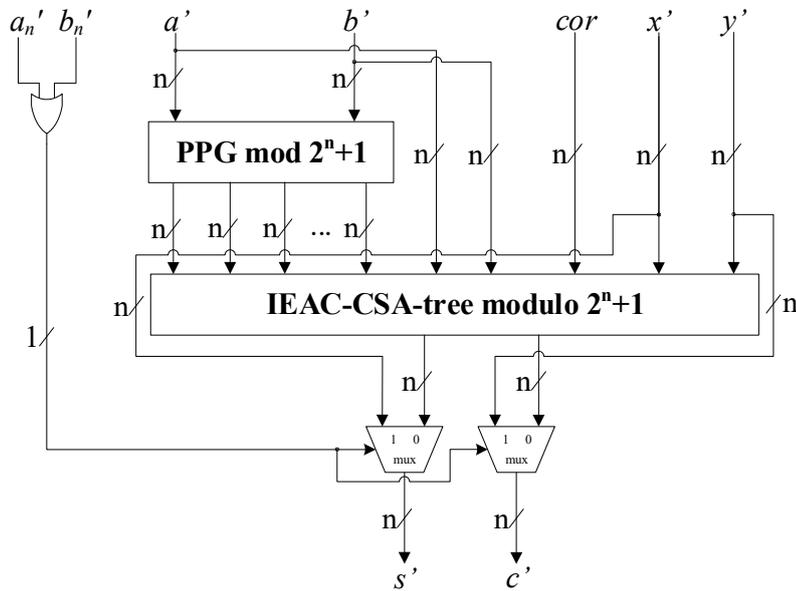


Рис. 8. Архитектура блока IEAC-TMAC

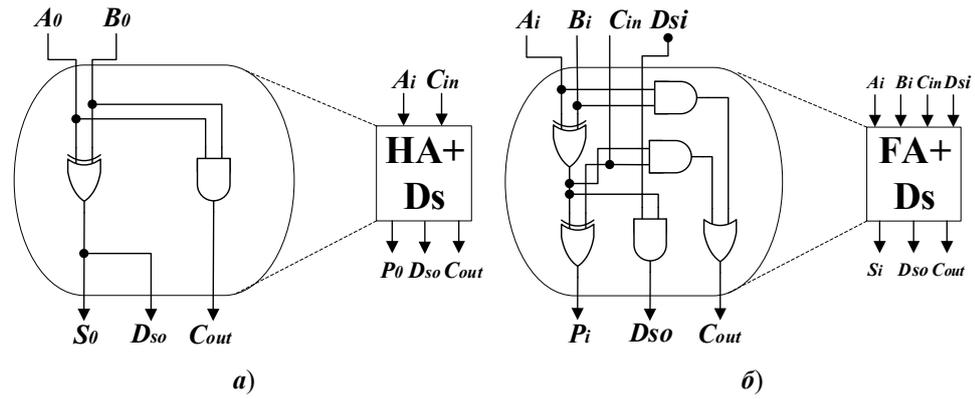


Рис. 9. Архитектуры сумматоров с учетом бита суммы:  
*a* – полусумматор; *б* – полный сумматор

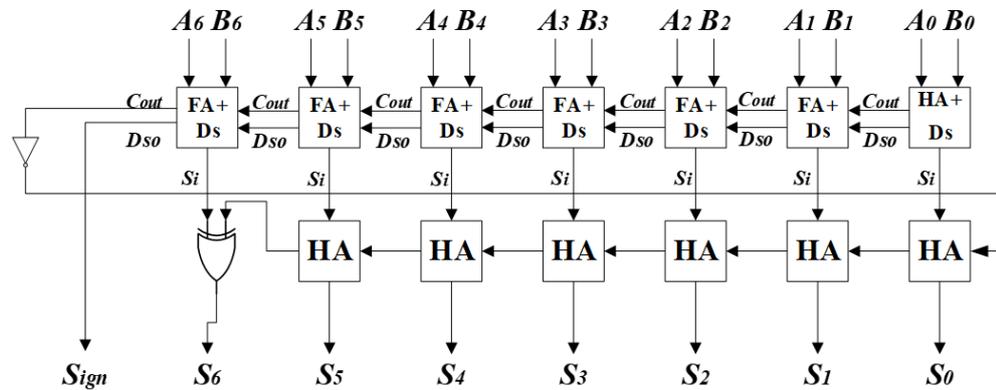


Рис. 10. Архитектура 7-битного сумматора  
 IEAC-CRA без учета определителя нуля

### 5. Аппаратное моделирование

Аппаратное моделирование проводилось на FPGA в среде Xilinx Vivado версии 2018.3. Цель данного моделирования заключается в подтверждении теоретического обоснования снижения аппаратных затрат с использованием модулей СОК вида  $\{2^n - 1, 2^n, 2^n + 1\}$  по сравнению с модулями  $\{2^n - 1, 2^n\}$ . Наборы модулей СОК составлялись из возможных комбинаций из 3–8 оснований с разрядностями  $2 \leq n \leq 12$ . Результаты моделирования, представленные в табл. 1, показывают, что фильтрация в СОК с использованием модулей вида  $\{2^n - 1, 2^n\}$  позволяет получить низкую задержку по сравнению с модулями  $\{2^n - 1, 2^n, 2^n + 1\}$  на 9,1–16,4 %, но набор модулей  $\{2^n - 1, 2^n, 2^n + 1\}$  продемонстрировал низкие показатели по аппаратным затратам по сравнению с фильтром на модулях  $\{2^n - 1, 2^n\}$  на 22,9–32,5 % в зависимости от порядка фильтра.

Таблица 1

## Результаты моделирования фильтров в СОК

Порядок фильтра	3 порядок		7 порядок		15 порядок		31 порядок		
	Площадь, LUTs	Задержка, нс	Площадь, LUTs	Задержка, нс	Площадь, LUTs	Задержка, нс	Площадь, LUTs	Задержка, нс	
Набор модулей СОК	(2047, 2048, 4095)	1290	30,329	2026	46,848	3752	83,923	8765	144,681
	(127, 255, 511, 512)	1014	<b>25,783</b>	1621	<b>41,534</b>	3518	72,361	8037	129,407
	(31, 32, 127, 255, 511)	1047	27,116	1665	43,225	3816	<b>71,073</b>	6578	<b>124,031</b>
	(7, 15, 16, 31, 127, 2047)	923	29,019	1442	46,178	3275	87,874	6696	152,175
Наборы с модулями $(2^n, 2^n - 1)$	(1025, 2047, 2048)	1463	36,610	2566	51,931	5263	88,005	13032	167,468
	(129, 257, 511, 512)	1054	33,142	1615	51,324	3309	76,850	8368	164,392
	(255, 257, 511, 512)	1129	32,319	1774	50,884	3667	78,582	9613	159,347
	(33, 65, 127, 128, 257)	1256	31,417	2147	51,722	3972	89,503	11477	161,229
	(63, 65, 127, 128, 257)	1134	33,376	1936	51,533	3689	84,969	10865	160,424
	(17, 31, 33, 64, 65, 127)	896	28,782	1582	47,564	3456	81,113	6878	135,126
	(7, 9, 17, 31, 32, 65, 127)	<b>623</b>	28,398	<b>1109</b>	47,600	<b>2526</b>	82,741	<b>4941</b>	135,369
	(3, 5, 7, 8, 17, 31, 127, 257)	818	32,826	1251	52,028	3752	83,923	5859	166,595

### Заключение

Предложен подход к реализации цифровой фильтрации сигналов в СОК с наборами модулей вида  $\{2^n - 1, 2^n, 2^n + 1\}$ . Результаты аппаратного моделирования показали, что фильтры с данными модулями занимают меньше аппаратных затрат по сравнению с набором модулей  $\{2^n - 1, 2^n\}$  до 32,5 %, но при этом повысилась задержка до 16,4 %.

Разработанные фильтры можно использовать для улучшения технических характеристик цифровых устройств, в которых главным приоритетом является минимизация аппаратных затрат и низкие показатели энергопотребления. Будущие исследования будут направлены на внедрение предложенных фильтров в устройства цифровой обработки сигналов для решения задач машинного обучения, обработки изображений, обработки звука и пр.

### Список литературы

1. Sundararajan D. D. *Digital Signal Processing*. Cham : Springer International Publishing, 2021. 399 p.
2. Chandra A., Chattopadhyay S. Design of hardware efficient FIR filter: A review of the state-of-the-art approaches // *Engineering Science and Technology, an International Journal*. 2016. Vol. 19, № 1. P. 212–226.
3. Alwahab D. A., Zaghar D. R., Laki S. FIR Filter Design Based Neural Network // 2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP). 2018. P. 1–4.
4. Suresh S., Lal S. Two-Dimensional CS Adaptive FIR Wiener Filtering Algorithm for the Denoising of Satellite Images // *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 2017. Vol. 10, № 12. P. 5245–5257.
5. Madouri Z. B., Said N. H., Ali Pacha A. Image encryption algorithm based on digital filters controlled by 2D robust chaotic map // *Optik*. 2022. Vol. 264. P. 169382. doi:10.1016/j.ijleo.2022.169382
6. Акушский И. Я., Юдицкий Д. И. *Машинная арифметика в остаточных классах*. М. : Сов. Радио, 1968. 440 с.
7. Исупов К. С. Методика выполнения базовых немодульных операций в модулярной арифметике с применением интервальных позиционных характеристик // *Известия высших учебных заведений. Поволжский регион. Технические науки*. 2013. № 3. С. 26–39.
8. Lyakhov P., Valueva M., Valuev G., Nagornov N. High-Performance Digital Filtering on Truncated Multiply-Accumulate Units in the Residue Number System // *IEEE Access*. 2020. Vol. 8. P. 209181–209190.
9. Jaberipur G., Belghadr A., Nejati S. Impact of diminished-1 encoding on residue number systems arithmetic units and converters // *Computers & Electrical Engineering*. 2019. Vol. 75. P. 61–76.
10. Parhami B. *Computer Arithmetic: Algorithms and Hardware Designs*. London, U.K. : Oxford Univ. Press, 2010. 492 p.
11. Patel B. K., Kanung J. Diminished-1 multiplier using modulo adder // *International Journal of Engineering & Technology*. 2018. Vol. 7. P. 31–35.

### References

1. Sundararajan D.D. *Digital Signal Processing*. Cham: Springer International Publishing, 2021:399.
2. Chandra A., Chattopadhyay S. Design of hardware efficient FIR filter: A review of the state-of-the-art approaches. *Engineering Science and Technology, an International Journal*. 2016;19(1):212–226.

3. Alwahab D.A., Zaghar D.R., Laki S. FIR Filter Design Based Neural Network. *2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*. 2018:1–4.
4. Suresh S., Lal S. Two-Dimensional CS Adaptive FIR Wiener Filtering Algorithm for the Denoising of Satellite Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 2017;10(12):5245–5257.
5. Madouri Z.B., Said N. Hadj, Ali Pacha A. Image encryption algorithm based on digital filters controlled by 2D robust chaotic map. *Optik*. 2022;264:169382. doi:10.1016/j.ijleo.2022.169382
6. Akushskiy I.Ya., Yuditskiy D.I. *Mashinnaya arifmetika v ostatochnykh klassakh = Machine arithmetic in residual classes*. Moscow: Sov. Radio, 1968:440. (In Russ.)
7. Isupov K.S. A technique for performing basic non-modular operations in modular arithmetic using interval positional characteristics. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki = University proceedings. Volga region. Engineering sciences*. 2013;(3):26–39. (In Russ.)
8. Lyakhov P., Valueva M., Valuev G., Nagornov N. High-Performance Digital Filtering on Truncated Multiply-Accumulate Units in the Residue Number System. *IEEE Access*. 2020;8:209181–209190.
9. Jaberipur G., Belghadr A., Nejati S. Impact of diminished-1 encoding on residue number systems arithmetic units and converters. *Computers & Electrical Engineering*. 2019;75:61–76.
10. Parhami B. *Computer Arithmetic: Algorithms and Hardware Designs*. London, U.K.: Oxford Univ. Press, 2010:492.
11. Patel B.K., Kanung J. Diminished-1 multiplier using modulo adder. *International Journal of Engineering & Technology*. 2018;7:31–35.

#### Информация об авторах / Information about the authors

**Максим Валерьевич Бергерман**

младший научный сотрудник,  
Северо-Кавказский центр  
математических исследований  
(Россия, г. Ставрополь, ул. Пушкина, 1)

E-mail: maxx07051997@inbox.ru

**Maksim V. Bergerman**

Junior researcher, North Caucasus  
Center for Mathematical Research  
(1 Pushkina street, Stavropol, Russia)

**Автор заявляет об отсутствии конфликта интересов / The author declares no conflicts of interests.**

**Поступила в редакцию / Received 29.12.2022**

**Поступила после рецензирования и доработки / Revised 14.02.2023**

**Принята к публикации / Accepted 10.03.2023**