

**А.К. МАИЛЯН, А.Ж. МОМДЖЯН**

**ИССЛЕДОВАНИЕ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ ТЕХНОЛОГИЙ  
WEB ПРОГРАММИРОВАНИЯ В СРЕДЕ JAVA**

Исследованы технологии Web программирования в среде Java. Проведен сравнительный анализ по следующим параметрам: быстрое прототипное приложение, простота в использовании, документация и обратная связь, пропускная способность и масштабируемость.

**Ключевые слова:** прототип, проект, инструмент, шаблон, инкапсуляция, интерфейс, ссылка, пропускная способность, класс, конфигурация.

**A.K. MAYILYAN, A.J. MOMJIAN**

**RESEARCH AND COMPARATIVE ANALYSIS OF WEB  
TECHNOLOGIES OF PROGRAMMING IN JAVA**

A research has been carried out on web development technologies in Java environment, and a comparative analysis has been performed according to the following parameters: rapid application prototyping, ease of use, documentation and community, throughput and scalability.

**Keywords:** prototype, project, tool, template, encapsulation, interface, link, bandwidth, class, configuration.

УДК 681.3

**А.К. ТУМАНЯН, А.А. АМАЗАСПЯН, Э.В. ВИРАБЯН**

**ПРИМЕНЕНИЕ КОНВЕЙЕРА ДЛЯ ВЫПОЛНЕНИЯ ОПЕРАЦИЙ  
ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ В СИСТЕМЕ ОСТАТОЧНЫХ  
КЛАССОВ**

Статья посвящена применению вычислений в системе остаточных классов (СОК) для решения задач цифровой обработки сигналов. Предложен вариант применения конвейера для реализации основной операции цифровой обработки сигналов Multiply and Accumulate (MAC) в СОК. Приводятся структура конвейера и алгоритм его работы. Для управления конвейером использован счетчик Джонсона.

**Ключевые слова:** конвейер, MAC-операция, система остаточных классов, модулярное умножение.

Одним из примеров использования СОК являются системы цифровой обработки сигналов. Система остаточных классов показала себя как выгодный инструмент повышения эффективности цифровых фильтров.

Для любой системы взаимно простых чисел  $p_1, \dots, p_n$  любое число  $X$  из диапазона  $[0; M-1)$ , где  $M = p_1 * p_2 * \dots * p_n$ , взаимно однозначно представимо в виде вектора  $(x_1, x_2, \dots, x_n)$ , где  $x_i = X \bmod p_i$ ,  $x_1, x_2, \dots, x_n$  – остатки (вычеты) числа по заданной системе модулей.

Основное свойство СОК – снижение разрядности. Для вычисления суммы или произведения двух чисел, представленных в СОК, достаточно сложить и умножить соответствующие остатки-разряды СОК. Операции сложения и вычитания в СОК из-за ограниченной разрядности модулей требуют меньше времени на распространение переноса. Так как модули не зависят друг от друга, операции могут выполняться параллельно. Иными словами, помимо снижения разрядности, добавляется параллелизм уровня модулей, который принципиально невозможен в позиционной системе [1].

Основной проблемой модулярной арифметики и, в частности, СОК является существование так называемых немодульных операций. К такому классу операций относятся, например, сравнение и деление чисел. Данные операции имеют позиционную природу, поэтому для их выполнения требуется восстановление позиционного представления чисел.

Цифровая обработка - арифметическая обработка в реальном масштабе времени последовательности значений амплитуды сигнала, получаемых через равные промежутки времени. Задачи цифровой обработки сигналов: фильтрация сигнала, свертка двух сигналов (смещение), вычисление значений корреляционной функции двух сигналов, усиление, ограничение или трансформация сигнала, прямое/обратное преобразование Фурье [2].

Схема модулярной МАС операции представлена на рис. 1.

Количество тактов устройства, которое требуется для выполнения операции МАС, в основном зависит от того, сколько тактов требуется для выполнения операции умножения.

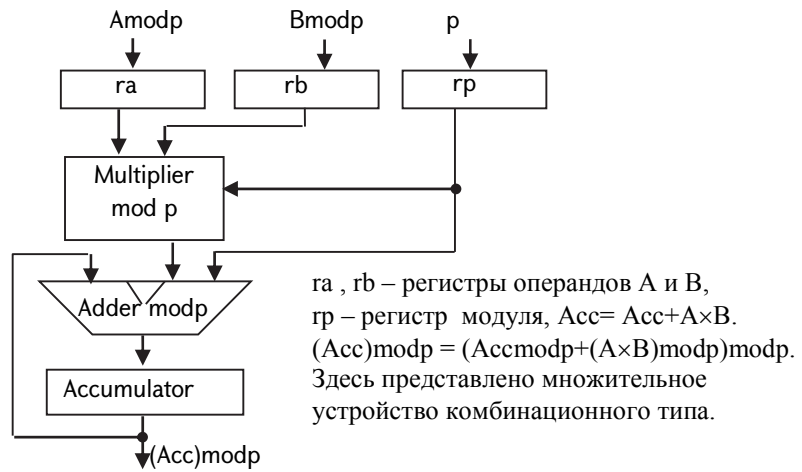


Рис. 1. Схема блока модулярной MAC операции

**Последовательный умножитель.** При последовательном умножении число тактов равно числу разрядов операндов, которое определяется разрядностью модуля  $p$  (обозначим  $n$ ). Сигнал в каждом такте проходит через комбинационную схему, которая содержит один сумматор и один мультиплексор. В модулярном умножителе сигнал в каждом такте проходит через три сумматора и три мультиплексора (рассматривается сумматор с последовательным переносом) [3].

Определим максимальную длительность такта  $T1$ :  $T1_{max} = T_{reg} + T_{pd\_log}$ , где  $T_{reg}$  – задержка установления информации на выходе регистра,  $T_{pd\_log}$  – максимальная задержка распространения сигнала через комбинационную схему в течение одного такта;  $T_{pd\_log} = 3T_{pd\_add} + 3T_{pd\_mux} = 3(n+1) + 3 = 3n+6$ , где  $(n+1)$  – число разрядов сумматоров, через которые проходит сигнал переноса в наихудшем случае (один бит добавляется для знака). Для упрощения можно считать, что  $T_{pd\_mux} = 1$ . 3 – число пар сумматор (вычитатель) - мультиплексор. Так как умножение осуществляется за  $n$  тактов, то общее время умножения будет  $(3n+6)n = 3n^2+6n$ .

Так как числа поступают на вход потоком, то число тактов нужно умножить на длину (количество чисел) входного потока данных (например,  $N$ ). Тогда общее число тактов работы схемы будет равно  $K1 = N(3n^2+6n)$ .

**Комбинационный умножитель.** Схема комбинационного модулярного умножителя сложнее, чем схема последовательного умножителя. Число сумматоров немодулярного  $n$ -разрядного матричного умножителя -  $(n-1)$ . Разрядность -  $(n+1)$ . Сигнал переноса распространяется в наихудшем случае через  $3(n+1)-4=3n-1$  сумматоров. В модулярном комбинационном умножителе

число сумматоров будет в два раза больше, т.е.  $2(n-1)$  сумматора с разрядностью  $(n+1)$  для умножения на каждый разряд множителя. Кроме этого, добавляется по одному сумматору и мультиплексору для вычисления модуля сдвигаемого влево множимого. Сигнал переноса в наихудшем случае распространяется через  $3 \cdot (2n+2) - 4$  сумматора.

$$T_{2\max} = T_{pd\_log} = (6n+2) + 3pd\_mux = 6n+5.$$

Число тактов работы схемы равно  $K_2 = 3 \cdot N$  при длительности такта  $T_{2\max}$ .

**Устройство умножения конвейерного типа.** Как известно, наиболее эффективным методом повышения производительности систем является конвейерный подход. Рассмотрим, как можно использовать преимущества конвейера для выполнения MAC-операции. В статье предлагается на каждой стадии производить умножение на один разряд.

Число итераций операции умножения равно  $n$ . В каждом такте к сумме частичных произведений добавляется очередное частичное произведение. Конвейер состоит из трех этапов: Prolog, Kernel и Epilog.

Когда конвейер находится на этапе Prolog, на конвейер запускаются новые итерации, но с конвейера не сходит ни одна итерация, то есть еще нет результатов. Когда конвейер находится в стадии Kernel, в каждом такте работы конвейера одна итерация заходит на конвейер, и одна итерация сходит с конвейера. Когда конвейер находится в стадии Epilog, новые итерации на конвейер не поступают, а с конвейера сходят результаты последних итераций.

**Разработка конвейера MAC-операции.** Модули  $p$  принимают значения до 31 (рассмотрим построение конвейера для модуля  $p$ , содержащего  $n=5$  разрядов). На каждой стадии конвейера происходит умножение на 1 бит множителя (регистр  $B$ ). Конвейер содержит 5 стадий операции умножения. На каждой стадии происходит умножение операнда  $A$  на один разряд  $B$ . Числа  $A$  и  $B$  поступают на конвейер по модулю  $p$ .

Умножение производится со сдвигом множимого влево, а множителя вправо на 1 разряд.  $ra$  – регистр множимого,  $rb$  – регистр множителя.

Схема одной стадии конвейера приведена на рис.2. На каждой итерации происходит следующее:  $SM1$  и  $MUX1$  формируют  $(2ra) \bmod p$ ;  $MUX2$ ,  $SM2$ ,  $SM3$  и  $MUX3$  формируют сумму частичных произведений по  $\bmod p$ .

Очередное частичное произведение, которое добавляется к содержимому аккумулятора, определяется следующим выражением:

$$acc = ((2ra \cdot rb[0]) \bmod p + acc) \bmod p.$$

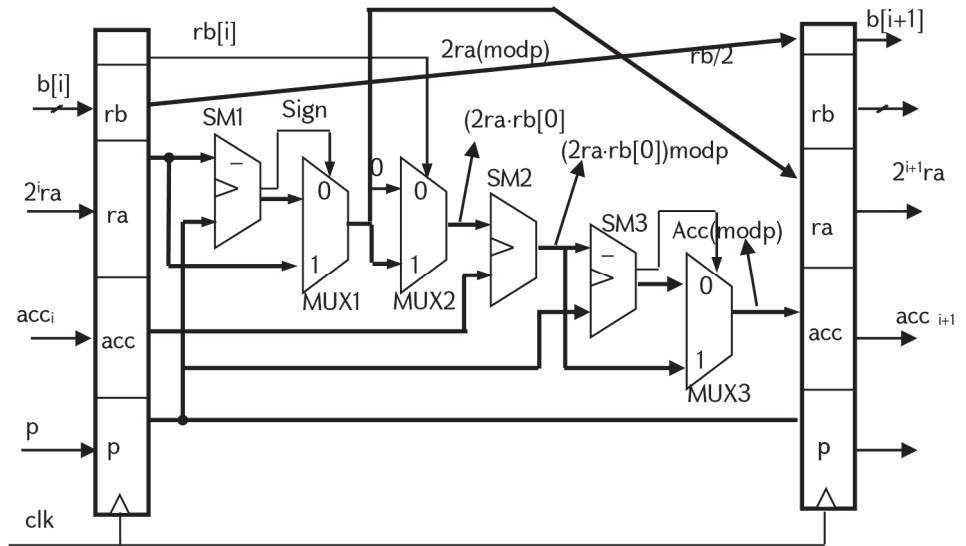


Рис. 2. Схема одной стадии конвейера

На рис.3 представлен 5-ступенчатый синхронный конвейер. Стадии конвейера разделены конвейерными регистрами R0, R1, R2, R3, R4. Управление конвейером осуществляет FSM.

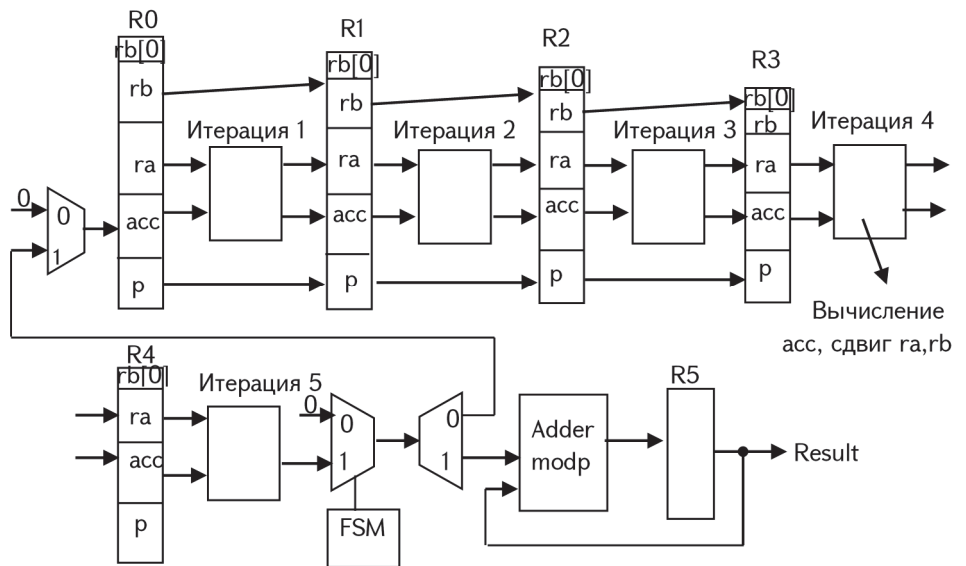


Рис. 3. Схема конвейера

При занесении информации с  $i$ -й стадии конвейера на  $(i+1)$  стадию га сдвигается влево на один бит, а  $rb$  сдвигается на 1 бит вправо. При этом на

каждой следующей стадии разрядность поля  $gb$  уменьшается на 1 бит. На последней пятой стадии  $gb$  содержит один разряд.

В каждом такте работы конвейера на вход конвейера поступает пара чисел  $a_i, b_i$ , начиная с  $a_0, b_0$ , затем  $a_1, b_1, a_2, b_2, \dots, a_{n-1}, b_{n-1}$ . По фронту синхросигнала информация на конвейере продвигается на одну стадию вперед.

В R5 записывается информация из асс, когда конвейер переходит в стадию Epilog. В этой стадии на конвейер не передаются новые данные, а с содержимым R5 суммируются находящиеся на стадии Prolog последние 5 произведений  $a_i \cdot b_i$ :  $a_{n-5} \cdot b_{n-5}, a_{n-4} \cdot b_{n-4}, a_{n-3} \cdot b_{n-3}, a_{n-2} \cdot b_{n-2}, a_{n-1} \cdot b_{n-1}$ .

Для управления работой конвейера удобно использовать счетчик Джонсона. Число разрядов счетчика Джонсона должно быть равно числу стадий конвейера (для рассмотренного примера - 6).

Таблица 1

	q0 q1 q2 q3 q4 q5		
	0 0 0 0 0 0		
1	1 0 0 0 0 0	En0	Разрешение R0
2	1 1 0 0 0 0	En0, En1	Разрешение R0,R1
3	1 1 1 0 0 0	En0, En1, En2	Разрешение R0,R1,R2
4	1 1 1 1 0 0	En0, En1, En2, En3	Разрешение R0,R1,R2,R3
5	1 1 1 1 1 0	En0, En1, En2, En3, En4	Разрешение R0,R1,R2,R3,R4
6	1 1 1 1 1 0	En0, En1, En2, En3, En4	Разрешение R0,R1,R2,R3,R4
7	1 1 1 1 1 0	En0, En1, En2, En3, En4	Разрешение R0,R1,R2,R3,R4
8	1 1 1 1 1 0	En0, En1, En2, En3, En4	Разрешение R0,R1,R2,R3,R4
9	1 1 1 1 1 0	En0, En1, En2, En3, En4	Разрешение R0,R1,R2,R3,R4
10	1 1 1 1 1 0	En0, En1, En2, En3, En4	Разрешение R0,R1,R2,R3,R4
11	1 1 1 1 1 0	En0, En1, En2, En3, En4	Разрешение R0,R1,R2,R3,R4
12	1 1 1 1 1 0	En0, En1, En2, En3, En4	Разрешение R0,R1,R2,R3,R4
13	0 1 1 1 1 1	En1, En2, En3, En4, En5	Разрешение R1,R2,R3,R4,R5
14	0 0 1 1 1 1	En2, En3, En4, En5	Разрешение R2,R3,R4,R5
15	0 0 0 1 1 1	En3, En4, En5	Разрешение R3,R4,R5
16	0 0 0 0 1 1	En4, En5	Разрешение R4,R5
17	0 0 0 0 0 1	En5	Разрешение R5

Стадии 1, 2, 3, 4 – Prolog, стадии 5 ÷ 13 – Kernel, стадии 14 ÷ 16 – Epilog.

В табл.1 представлена схема управления сигналами разрешения  $En_0, En_1, En_2, En_3, En_4, En_5$  стадий конвейера. Сигналами разрешения являются выходы  $q_0, q_1, q_2, q_3, q_4, q_5$  счетчика Джонсона.

Число тактов работы конвейера равно  $N+n$  ( $N$  - длина входного потока данных).

На рис. 4 представлена блок-схема алгоритма управления счетчиком Джонсона.

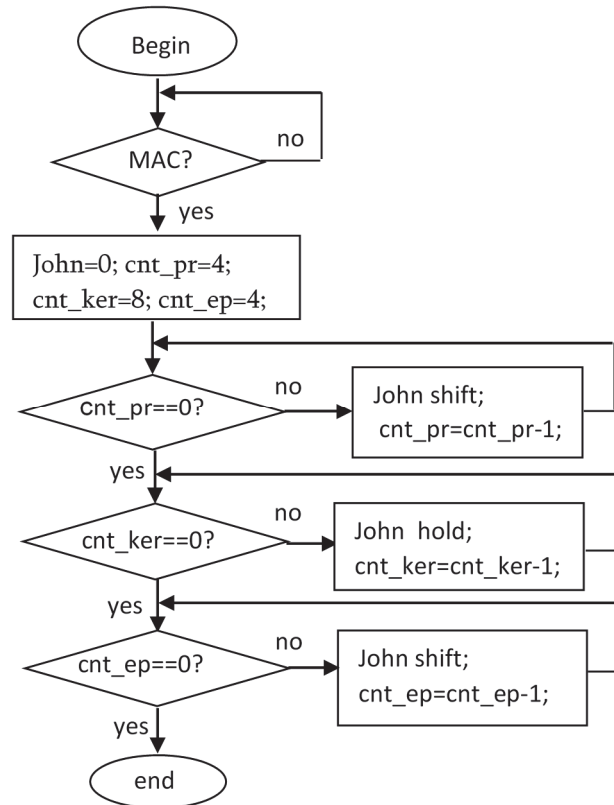


Рис. 4. Блок-схема алгоритма работы конвейера

Данные для сравнения трех рассмотренных вариантов схем реализации операции MAC приведены в табл. 2.

Таблица 2

Реализация	Число тактов	Число (n+1)-разрядных сумматоров на пути сигнала переноса	Число мультиплексоров
Конвейер	$N+n$	3	3
Последовательный умножитель	$3N(n^2+1)$	3	3
Комбинационный умножитель	$3N$	6	3

**Заключение.** Время выполнения определяется не только числом тактов, но и длительностью такта. Длительность такта для схем умножения с конвейером и последовательного умножителя совпадает и пропорциональна  $3n$ . Самую большую длительность такта имеет схема с комбинационным умножителем - пропорциональна  $6n$ . Длительность такта в схеме на основе комбинационного устройства умножения в два раза больше, чем в схеме на конвейере.

Наихудшим решением с точки зрения быстродействия является схема с последовательным умножителем, наилучшим – схема с конвейером.

По количеству  $(n+1)$ -разрядных сумматоров порядок следующий: схема комбинационного умножения содержит  $6n$  сумматоров, с конвейером -  $3n$  сумматора, схема последовательного умножителя содержит 3 сумматора.

Из известных схем конвейерного типа была рассмотрена схема DSP (Digital Signal Processing), используемая в ПЛИС 7-й серии фирмы Xilinx [4].

Конвейер DSP состоит из двух стадий. На первой стадии формируются два частичных произведения, а на следующей стадии они складываются между собой. Сложности этих реализаций не сопоставимы, так как разрядность чисел DSP - 18 бит, а результат - 36 бит. Достоинством вычислений в СОК является то, что сомножители и аккумулятор имеют одинаковую длину, так как операции сложения и умножения модулярные.

Исходя из вышеизложенного, следует, что предложенная конвейерная реализация операции MAC в СОК обеспечивает выигрыш по числу тактов по сравнению с реализацией без конвейера  $3N(n^2+1)/(N+n)$ . При этом сложность схемы возрастает в  $n$  раз.

#### СПИСОК ЛИТЕРАТУРЫ

1. К вопросу применения системы остаточных классов в современных устройствах цифровой обработки сигналов / Д.В. Юрданов, Д.В. Горденко, Н.В. Горденко и др. // Фундаментальные исследования. – 2016. – № 2-2. – С. 318-322.
2. Ричард Лайонс. Цифровая обработка сигналов. - М., 2006.
3. Туманян А.К., Амазаспян А.А., Костанян А.А. Выполнение операций умножения и деления в системе остаточных классов // Вестник НПУА: Сборник научных статей. - 2017. - №1.- С.109-116.
4. 7 Series DSP 48E1 User Guide, 2018.

**Ա.Կ. ԹՈՒՄԱՆՅԱՆ, Ա.Հ. ՀԱՄԱԶԱՍՊՅԱՆ, Է.Վ. ՎԻՐԱԲՅԱՆ**  
**ՄՆԱՅՈՐԴԱՅԻՆ ԴԱՍԵՐԻ ՀԱՄԱԿԱՐԳՈՒՄ ԱԶԴԱՆՇԱՆՆԵՐԻ ԹՎԱՅԻՆ**  
**ՄՇԱԿՄԱՆ ԳՈՐԾՈՂՈՒԹՅՈՒՆՆԵՐԻ ԿԱՏԱՐՈՒՄԸ ԿՈՆՎԵՅԵՐԻ**  
**ԿԻՐԱՌՄԱՄԲ**

Աշխատանքը նվիրված է մնացորդային դասերի համակարգում (ՄԴՀ) հաշվարկների կիրառմանը՝ ազդանշանների թվային մշակման խնդիրների լուծման համար: Առաջարկված է մնացորդային դասերի համակարգում ազդանշանների թվային մշակման հիմնական Multiply and Accumulate (MAC) գործողության իրագործման համար կոնվեյերի կիրառման տարբերակը: Ներկայացված են կոնվեյերի կառուցվածքը և աշխատանքի ալգորիթմը: Կոնվեյերի ղեկավարման համար օգտագործված է Ջոնսոնի հաշվիչը:

**Առանցքային բառեր.** կոնվեյեր, MAC-գործողություն, մնացորդային դասերի համակարգ, մոդուլյար բազմապատկում:

**A.K. TUMANYAN, A.H. HAMAZASPYAN, E.V. VIRABYAN**  
**THE USE OF THE PIPELINE TO PERFORM DIGITAL SIGNAL**  
**PROCESSING OPERATIONS IN THE RESIDUE NUMBER SYSTEM**

The paper is devoted to applying calculations in the system of residual classes (SRC) to solve tasks of the digital signal processing. A variant of using a pipeline for the implementation of the main operation of digital signal processing Multiply and Accumulate (MAC) in the system of residual classes is proposed. The structure of the pipeline and the algorithm of its operation are given. To control the pipeline Johnson's counter is used.

**Keywords:** pipeline, MAC-operation, modular multiplication.