

Д. т. н. С. А. НЕСТЕРЕНКО, к. т. н. О. Н. ПАУЛИН

Дата поступления в редакцию  
09.07 2007 г.

Украина, Одесский национальный политехнический университет  
E-mail: as\_nesterenko@ukr.net, paulin@te.net.ua

Оппонент к. т. н. Ю. П. МАРДАМШИИ  
(НКБ ЦОС ЮФУ, г. Таганрог)

## ПОСТРОЕНИЕ ОБОБЩЕННОЙ МОДЕЛИ ОПЕРАЦИИ СВЕРТКИ МНОГОРЯДНЫХ КОДОВ ПРИ ЦИФРОВОЙ ОБРАБОТКЕ СИГНАЛОВ

*Рассматриваются известные модели, процедуры и методы свертки многорядных кодов (МРК). Строится обобщенная модель операции свертки МРК с учетом переносов.*

Цифровая обработка сигналов (ЦОС) охватывает широкий круг практических приложений [1]: цифровая фильтрация, распознавание и кодирование речи, цифровая звукотехника, обработка изображений, быстрое преобразование Фурье и др. Как в этих, так и в других случаях ЦОС имеет общие особенности — большой объем вычислений и работу в реальном времени. Поэтому скорость обработки цифровых данных является во многих случаях определяющим фактором ЦОС.

Для аппаратной реализации ЦОС чаще всего используются сигнальные процессоры типа TMS320 (Texas Instruments), которые позволяют проводить обработку данных с высокой производительностью. Однако требования к быстродействию устройств ЦОС постоянно растут, поэтому актуальным является поиск новых подходов к построению таких устройств.

На рис. 1 представлена структура (а) цифрового фильтра (ЦФ) с бесконечной импульсной характеристикой (БИХ); структура содержит  $k$  параллельно включенных универсальных звеньев второго порядка, называемых биквадратными блоками (б).

Функционирование биквадратного блока может быть описано выражением

$$Y(n) = a_0(X(n) - b_1X(n-1) - b_2X(n-2)) + a_1X(n-1) + a_2X(n-2). \quad (1)$$

Из этого выражения видно, что оба сумматора, входной и выходной, оперируют с тремя операндами, т. е. являются трехоперандными. При умножении цифровых сигналов на коэффициенты  $a_i$  и  $b_j$  нужно одновременное суммирование частичных произведений (их групп или всех), т. е. множитель должен включать в себя многооперандный сумматор. Поскольку звенья БИХ включены параллельно (рис. 1, а), их выходные сигналы могут быть просуммированы одновременно. В этом случае число слагаемых (сигналов) равно  $k+1$ , т. е. сумматор также должен быть многооперандным. Множество операндов в совокупности можно рассматривать как

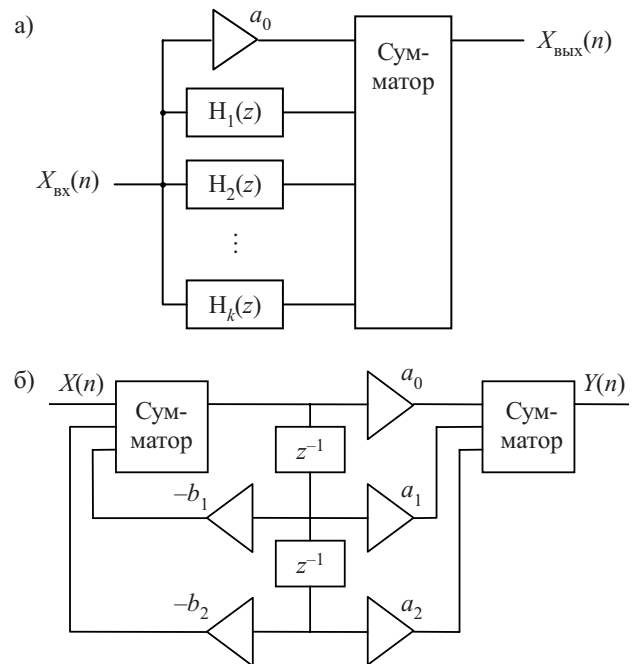


Рис. 1. Параллельная форма БИХ-фильтра: а — структура; б — биквадратный блок;  $H(z)$  — передаточная функция звена;  $z$  — параметр Z-преобразования

многорядные арифметические коды (МРК), которые нужно одновременно обработать.

Необходимость одновременной обработки многорядных кодов возникает также при суммировании потока данных, поступающих на вход суммирующего устройства, когда требуется за один операционный такт сложить группу слагаемых. Такая ситуация может возникнуть при решении задач матричной алгебры, при усреднении множества данных и т. п.

Для эффективной одновременной обработки множества данных в потоке, в том числе и при ЦОС, предложена *многооперандная* обработка [2], которая основана на следующей идее: любая арифметическая операция (например умножение) может быть сведена к операции сложения, которую можно представить как операцию свертки МРК. Такая обработка в зарубежной литературе носит название «распределенная арифметика».

Идея свертки МРК при выполнении операции сложения впервые была рассмотрена в [3] и развита в [4–8]; разработаны и соответствующие операци-

онные элементы [5, 9, 10]. В последнее время разработаны эффективные процедуры и методы свертки МРК [11—16], опирающиеся на аппарат симметрических булевых функций [17].

Из требования повышения скорости обработки потока данных вытекает необходимость ускорения операции свертки МРК и разработки подходящей для этого модели, которая учитывала бы все наработки в этой области.

Ниже рассматриваются известные процедуры и методы свертки МРК и организация переносов при этом; на основе обобщения свойств и особенностей рассмотренных методов строится усовершенствованная модель операции свертки МРК.

**Известные модели, процедуры и методы свертки МРК**

Рассмотрим известные модели операции суммирования в потоке данных [3, 7].

Заданы  $M$  слагаемых  $A_1, A_2, \dots, A_M$  в двоичной системе счисления (СС):

$$A_i = 2^{g_j} \cdot \sum_{i=1}^{k_j} 2^{i-1} a_{ji}, \quad (2)$$

где  $i$  — номер разряда;  
 $k_j$  — разрядность  $j$ -го слагаемого;  
 $a_{ji}$  — значение  $i$ -го разряда для  $j$ -го числа.

При этом слагаемые сдвинуты влево на  $g_j (g_j \geq 0)$  разрядов и некоторые из разрядов чисел  $A_j$  могут быть равны 0 [11].

В модели *горизонтальной обработки* искомая сумма (без учета входных переносов) равна

$$Q = \sum_{j=1}^M A_j = \sum_{j=1}^M 2^{g_j} \cdot \sum_{i=1}^{k_j} 2^{i-1} a_{ji}, \quad (3)$$

причем для суммирования используются бинарные сумматоры с горизонтальной организацией переноса по всей цепочке разрядов.

Перейдем к модели *вертикальной обработки*. Изменяя в формуле (3) порядок суммирования и заменяя  $k_j$  на  $N$ , а  $M$  на  $M_i$ , получим

$$Q = \sum_{i=1}^N 2^{i-1} \cdot \sum_{j=1}^{M_i} a_{j(i)}, \quad (4)$$

где  $N$  — количество разрядных срезов ( $N = \max(k_j + g_j), 1 \leq j \leq M$ );  
 $M_i$  — количество слагаемых в  $i$ -м разрядном срезе ( $1 \leq i \leq N$ ).

Отметим, что рядность суммируемого кода  $M' = \max M_i (1 \leq i \leq N)$ .

Обозначим  $a_i = \sum_{j=1}^{M_i} a_{j(i)}$ , где  $a_i$  — количество единиц в  $i$ -м разрядном срезе. Тогда из (4) следует, что

$$Q = \sum_{i=1}^N 2^{i-1} a_i. \quad (5)$$

Обозначим  $M'$  минимальное количество разрядов, необходимое для представления поразрядной

суммы  $\sum_{j=1}^{M_i} a_{j(i)}$ . Поскольку эта сумма целочисленна, то из выражения

$$\sum_{j=1}^M a_{j(i)} \leq M \quad (6)$$

следует, что  $M'$  — наименьшее число, которое для двоичной системы счисления ( $\alpha=2$ ) удовлетворяет неравенству [3, 7]

$$M \leq 2^{M'} - 1; \quad (7)$$

его решением является

$$M' = \lceil \log_2(M+1) \rceil, \quad (8)$$

где  $\lceil s \rceil$  — наименьшее целое число, не меньшее, чем  $s$ .

Отметим, что описание моделей вертикальной обработки (выражения (4)—(8)) может быть обобщено на случай двоичнокодированных и недвоичных СС ( $\alpha > 2, \alpha$  — целое).

В моделях (2) и (4) представлено поразрядное суммирование битов слагаемых, но не показаны ни порядок суммирования, ни организация переносов; эти моменты должны особо оговариваться.

Существуют два вида обработки (свертки) МРК — горизонтальная и вертикальная.

Первоначально процедура свертки кодов включала в себя поэтапное вычисление парных сумм на основе *горизонтальной* обработки, при которой использовалась бинарная операция поразрядного суммирования с полным распространением всех горизонтальных переносов. В этом случае количество этапов свертки равно  $\lceil \log_2 n \rceil$ , где  $n$  — количество разрядов слагаемых. Однако многоэтапная процедура свертки МРК на основе бинарного сумматора требует большого времени и ухудшает производительность устройства свертки (компрессора) в целом.

В настоящее время все шире применяется вертикальная обработка кодов. Процедуры, которые используются при этом, можно разделить на два направления — с непосредственной обработкой кодов и с предварительным упорядочиванием кодов.

Первое направление основано на поэтапной свертке  $m$ -рядного кода (группы данных одинаковой разрядности) в двухрядный код с последующим его суммированием на стандартном двухоперандном (бинарном) сумматоре с параллельным переносом (далее — параллельный сумматор). На каждом этапе количество полученных в результате рядов кода определяется по (8), так что процедура может быть представлена в виде  $m \rightarrow m' \rightarrow m'' \rightarrow \dots \rightarrow 2$ . При этом используется *метод*, основанный на базовой операции суммирования значений битов разрядного среза (РС) в едином операционном цикле. Для собственно свертки кодов используются одноразрядные полные сумматоры (как отдельные операционные элементы (ОЭ) либо как их сборки в виде узлов, деревьев), в каждом из которых складывается  $m$  цифр данного разряда и вычисляется их сумма в виде  $m'$ -разрядного числа, т. е. осуществляется преобразование  $m \rightarrow m'$ . Это направление представлено работами Храпченко [3, 4], Брика и Лушпина [5], Уоллеса [9], Санторо [10].

Второе направление развивалось в работах [6, 8]. В процедуре Ромма биты одного разряда результатов  $m'$  суммирования всех РС совмещаются (для каждого РС они помещаются в результирующей разрядной решетке по диагонали), затем полученный код результата добавляется к новой группе из  $m$  разрядов, и далее уже обрабатывается  $(m+m')$ -рядный код. Суммирование осуществляется по методу предварительного упорядочивания кодов и проводится на параллельном счетчике одnorазрядных чисел [18], построенном по принципу быстрого сдвига частичных кодов исходного двоичного непозиционного кода на каждом этапе его упорядочивания ( $k$  — номер этапа); результирующий код является унитарным. Отметим, что для подсчета числа единиц в исходном коде и представления его в двоичном виде упорядоченный код необходимо дополнительно пропустить через шифратор, а затем перевести в двоичный ( $\alpha$ -ичный) код.

Окончательное суммирование с учетом переносов предпоследнего шага осуществляется параллельным сумматором на последнем шаге.

Процедура, изложенная в [8], заключается в многократном использовании метода деления упорядоченного кода РС на выбранное основание системы счисления  $\alpha \geq 2$ .

Гамаюн [7] предложил комбинированную процедуру, указав на резкое снижение эффективности свертки (в смысле скорости) на последних ее этапах. Его идея состоит в том, что одnorядный код результата можно получить за три этапа, независимо от количества операндов исходного кода. На 1-м этапе аналогично [3] производится сжатие  $m$ -рядного кода в  $m'$ -рядный код ( $m' = \lceil \log_2(m-1) \rceil + 1$ ) с помощью массива многовходовых одnorазрядных сумматоров. На 2-м этапе проводится секционирование  $m'$ -рядного кода с целью его свертки в двухрядный код, причем каждая секция обрабатывается  $k$ -разрядным  $m'$ -операндным сумматором, где  $k = \lceil \log_2(m'-1) \rceil + 1$ . Разрядность секции  $k$  выбирается так, чтобы результат суммирования  $m'$   $k$ -разрядных чисел имел не более  $2k$  разрядов, что обеспечивает получение на выходе второго этапа двухрядного кода результата. Именно секционирование и упорядочивание кодов при суммировании позволяют добиться ускорения свертки МРК. На 3-м этапе двухрядный код суммируется стандартным образом на бинарном сумматоре с параллельным переносом.

Необходимость получения на втором этапе двухрядного кода ограничивает снизу минимальную разрядность секции  $k$ , что в зависимости от аппаратной реализации блока обработки секций приводит либо к чрезмерным затратам оборудования, либо к увеличению времени распространения сигнала и ухудшению быстродействия компрессора.

Методы вертикальной обработки можно разделить на два множества: 1) методы обработки кодов

каждого РС в отдельности; 2) методы обработки группы РС.

В первое множество входят:

— методы обработки РС одnorазрядными многооперандными сумматорами [3, 4]. Они подразумевают пирамидальную организацию при схемотехнической реализации и не отличаются достаточно высоким быстродействием схемы, т. к. требуют дополнительного времени на обработку переносов;

— методы вертикального сложения [6, 8], которые заключаются в предварительном упорядочивании кода РС. Схемы на основе этой группы методов основаны либо на параллельных счетчиках [6], либо на пирамиде делителей [8]; эти схемы не являются достаточно быстрыми, поскольку требуют больших затрат времени на предварительное упорядочивание исходного кода РС.

Второе множество методов включает в себя методы покрытия битов кодов деревьями Уоллеса [9] и Санторо [10]. Деревья в первом случае строятся на операционных элементах типа 6-2, а во втором — на ОЭ типов 4-2 и 8-2; в свою очередь эти ОЭ строятся на одnorазрядных полных сумматорах — CSA (carry-save adder), которые являются ОЭ типа 3-2.

Для этих методов характерна одновременная обработка всего слоя рядов кодов независимо от разрядности слоя за счет использования перетекания переносов ( $Cin_{i+1} = Cout_i$ ) от места формирования переноса данного разряда до выхода результата последующего разряда (Carry). Это является достоинством этих методов.

Нами показано, что имеет место следующая асимптотическая зависимость количества рангов  $r$  полной пирамиды от количества рядов  $m$  МРК при заданном коэффициенте свертки  $k$  (заданном ОЭ) и заданном количестве  $t$  обрабатываемых рядов на заключительном этапе процедуры свертки МРК ( $t=2$  для традиционных процедур и  $t=3$  — для модифицированной нами процедуры):

$$r = \log_k m/t; \tag{9}$$

скачкообразное изменение  $r$  происходит при добавлении к полной пирамиде одного ряда.

При повышении значения  $k$  ( $m = \text{const}$ ) глубина схемы уменьшается. Отметим к тому же, что при увеличении рядности  $m$  увеличивается не только  $r$ , но и (при многооперандной обработке) количество переносов, что усложняет схему и дополнительно увеличивает время обработки МРК.

Схемы на упомянутых деревьях тоже относятся к классу пирамидальных схем; глубина схемы определяется по (9), при этом  $k=2, 3, 4$ .

Отметим, что формула (9) не работает при дробном  $k$ , например, при  $k=3/2$  (для ОЭ типа 3-2), т. к. округление, к которому нужно прибегать при использовании (9), искажает значение  $r$ ; вместо аналитического выражения можно предложить таблицу.

Зависимость глубины схемы от количества свертываемых рядов МРК

$m$	3	4	5	6	7	8	9	10	11	12	13	14—19	20—28	29—42	43—63	64—94
$r$	1	2	3	3	4	4	4	5	5	5	5	6	7	8	9	10



К этому же множеству методов относятся методы, использующие исключение горизонтальных шагов [6]. В этих методах выполняются арифметические операции, сводящиеся к операции сложения, над потоком групп данных (группа — прямоугольный блок битов множества операндов определенной разрядности). Обработка осуществляется поблочно (на каждом шаге — свой блок), а обработка блока — поразрядно одновременно. Важная особенность обработки потока: переносы учитываются в конце выполнения всей операции, на заключительной стадии (отложенный перенос).

Время обработки РС  $T \approx (2\log_2 m + 3)\tau$  состоит из времени работы сдвигателя ( $2\tau \log_2 m$ ), времени шифрации результата обработки РС ( $2\tau$ ) и времени перевода сигнала шифратора в двоичный код ( $1\tau$ ). На заключительной стадии несколько рядов кодов предварительно сжимаются в два ряда, а далее используется бинарный полноразрядный параллельный сумматор. Это снижает эффект от экономии времени, который получен за счет исключения обработки переносов на промежуточных стадиях.

### Известные методы организации переносов

В операции свертки арифметических МРК важную роль играют методы организации межразрядных переносов [15]. Исследованием организации переноса занимались многие авторы. Так, Угрюмов [19] рассмотрел основные виды переносов при бинарном сложении, уделив много внимания групповым и многоярусным методам организации переноса. Глассер [20] предложил двухбитовую непозиционную организацию переноса для трехоперандного сумматора. Ромм [6] рассмотрел перенос, который можно назвать *отложенным*, при параллельном суммировании группы кодов. В деревьях Уоллеса и Санторо [9, 10] для свертки МРК используется перенос, который можно назвать *стекающим*.

Будем различать переносы поразрядные (внутренние) и результирующие (выходные) для всего компрессора (его каскада). Перенос образуется, если сумма битов операндов данного разряда превышает 1; количество битов переноса определяется максимальным значением суммы битов операндов данного РС.

Значение бита переноса определяется по тем же правилам, что и значение бита суммы, поскольку бит переноса определяется результатом суммирования битов операндов.

Переносы можно разделить на два вида: *позиционные* и *непозиционные*<sup>1</sup>. В свою очередь позиционные переносы делятся на *последовательные*, *сквозные*, *параллельные*, *стекающие* и др., а непозиционные — на переносы двух подвидов, *двухбитовые* и *многобитовые*. Наиболее эффективными являются параллельный позиционный и многобитовый непозиционный перенос; к последнему можно отнести также стекающий перенос. Особая организация переноса предложена Роммом.

Параллельный перенос в бинарном сумматоре организуется с помощью специальных функций ге-

нерации и прозрачности [19, 21]. При большой разрядности  $n$  сумматора она разбивается на группы по  $p$  ( $p \leq 8$ ) разрядов, и над всеми группами строится ярус ускоренного (параллельного) переноса; в свою очередь разрядность ( $k = n/p$ ) яруса также может быть разбита на свои группы, и тогда над ними строится следующий ярус схем ускоренного переноса. Важно отметить, что функции, описывающие перенос для ярусов, структурно одинаковы и подобны функциям генерации и прозрачности.

Идея организации переносов Ромма [6] сводится к следующему. На каждом шаге суммируются биты РС, составляющих группу данных. Переносы, возникающие при этом от каждого РС, запоминаются в разрядной решетке, причем они располагаются в ней по диагонали; суммирование битов переносов не производится. Эти биты присоединяются на следующем шаге к новой группе данных, и далее повторяется предыдущий шаг относительно расширенной совокупности битов. Так происходит до последнего шага, на котором производится окончательное суммирование на параллельном сумматоре с учетом битов переносов. Такого рода перенос будем называть *отложенным*. Достоинством этого подхода является то, что перенос может быть отложен до окончания серии арифметических операций, каждая из которых может быть выражена через операцию суммирования. Окончательное суммирование производится на бинарном сумматоре на быстрых сдвигателях [18].

Уоллес [9] и Санторо [10] предложили для свертки одного РС элементы типов 6-2 и 4-2, соответственно (рис. 2). Здесь действуют внутренние переносы ( $C_{out}$  из данного РС и  $C_{in}$  — в следующий РС). Образовавшиеся в данном разряде внутренние переносы являются входными битами в следующий разряд, где они формируют бит суммы следующего разряда и переносов через разряд — имеет место стекание переноса из верхнего яруса элемента данного разряда в нижний ярус элемента следующего разряда и, далее, стекание его на выход через разряд.

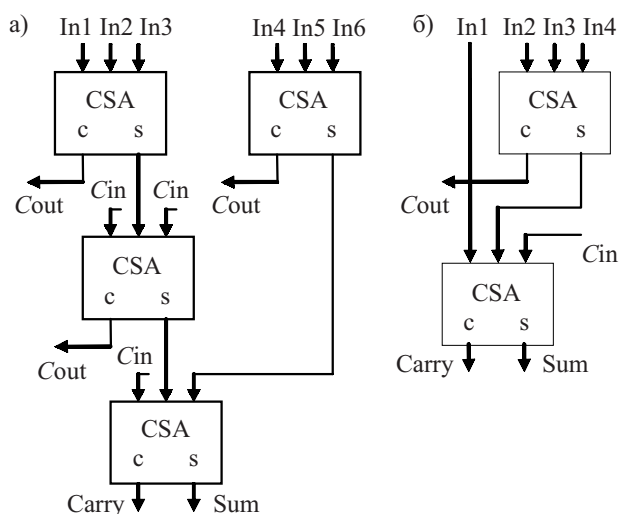


Рис. 2. Операционные элементы:

а — Уоллеса; б — Санторо;  $c, s$  — выходы переноса и суммы из сумматора CSA

<sup>1</sup> Связь между позиционным и непозиционным переносом рассмотрена в [12].

Принцип построения деревьев из таких ОЭ аналогичен предложенному Храпченко [3] и заключается в следующем: множество  $m$  битов данного РС обрабатывается параллельно элементами Уоллеса (Санторо) на данном этапе; полученные в результате двухрядные коды в совокупности ( $m'$ ) снова обрабатываются такими же ОЭ на следующем этапе, и так до тех пор, пока не останется один двухрядный код; окончательный результат получается при использовании бинарного параллельного сумматора — суммируются биты  $Sum_i$  и  $Carry_{i-1}$ .

В [5] рассмотрен пример четырехрядного узла 4-2, обрабатывающего 4 ряда кодов. Отметим, что для получения окончательного переноса из узла необходимо просуммировать переносы на 5-м разряде  $Cout$  и  $Carry$ , выходящих из старшего разряда.

ОЭ2 — это элементы, разработанные нами [22], подобие элементов Уоллеса и Санторо. Отличие заключается в том, что в ОЭ2 предусматривается несколько выходных переносов. Таковым элементом является разработанный нами [14] одноразрядный ОЭ типа 4-3 с двумя выходными переносами  $Carry$ ; этот компрессор введен в схему сумматора типа «ромб» с целью ее оптимизации по быстродействию. Несомненным достоинством деревьев при свертке МРК с их помощью является возможность обработки слоя кодов (4 или 6, а также кратное им число рядов) с практически неограниченной разрядностью. Однако результатующее сжатие 2-1 не является наилучшим решением. Лучшие результаты по быстродействию могут быть получены при заключительной свертке МРК вида 3-1 — см. выше.

Итак, рассмотренные виды переносов обладают определенными достоинствами и недостатками; выбор того или иного вида переносов зависит от конкретной ситуации (предъявленных требований по быстродействию и аппаратным затратам). По быстродействию их можно выстроить так: предпочтительнее подход Ромма, ему уступает подход Гамаюна, а далее идут с одинаковым быстродействием подходы Храпченко, Карцева и Брика, Санторо и Уоллеса.

**Обобщенная модель операции свертки МРК**

Анализ известных процедур и методов организации переносов выявил следующие их недостатки: узкая номенклатура конфигураций области битов (ОБ) (прямоугольник и параллелограмм); использование на последнем этапе свертки бинарного параллельного сумматора ( $t=2$ ); не уделяется достаточно внимания оптимальной организации переносов при свертке МРК.

На основе рассмотренных модели вертикальной обработки [7] (см. также (5)), процедур и методов свертки МРК нами предлагается обобщение модели операции свертки МРК, которое устраняет указанные недостатки.

На рис. 3 представлены предлагаемая модель операции свертки МРК и модели операционных элементов для свертки одного разряда (разрядного среза) МРК. Здесь обозначено:  $a_1, \dots, a_n$  — разрядные срезы (РС);  $l_1, \dots, l_n$  — количество битов (одноразряд-

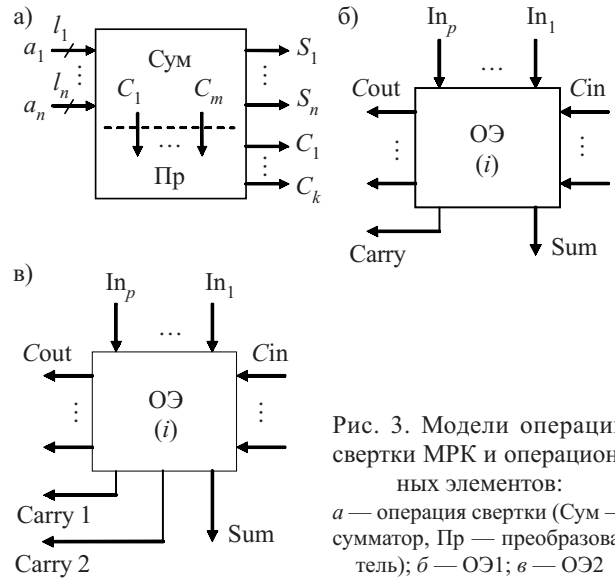


Рис. 3. Модели операции свертки МРК и операционных элементов:  
 а — операция свертки (Сум — сумматор, Пр — преобразователь); б — ОЭ1; в — ОЭ2

ных слагаемых) в РС;  $S_1, \dots, S_n$  — поразрядные значения суммы;  $C_1, \dots, C_m$  — внутренние переносы;  $C_1, \dots, C_k$  — результирующие позиционные переносы. Обозначения для ОЭ:  $In_1, \dots, In_p$  — биты РС;  $Cin$ ,  $Cout$  — биты внутреннего переноса, входного и выходного, соответственно;  $Carry$  — внешний перенос в следующий разряд;  $Sum$  — сумма битов РС.

Понятие «обобщенная модель операции свертки МРК» включает в себя следующие составляющие: 1) математическое описание операции свертки МРК (выражения (4)—(8)); 2) область битов операндов  $a_1, \dots, a_n$  произвольной разрядности; 3) способы разбиения исходной области битов на фрагменты; 4) процедуры свертки, включающие в себя несколько этапов, причем на каждом этапе используется свой метод обработки МРК; 5) методы организации переносов при свертке кодов; 6) модели ОЭ.

Под областью битов в общем случае понимается совокупность рядов арифметических кодов с произвольной разрядностью, по-разному сдвинутых друг относительно друга — см. (4). Для свертки МРК при большом числе рядов кодов используется разбиение множества рядов кодов на слои (их фрагменты) и свертка полученных слоев (их фрагментов), а далее — свертка рядов полученных результатов. Область битов может быть разбита на фрагменты [13], регулярные (например прямоугольники, ромбы и др.) и нерегулярные; в [13] же приведены теоремы о количестве переносов из данного фрагмента.

Необходимость разбиения определяется сложностью компрессора, которая в случае применения многооперандных структур пропорциональна квадрату числа разрядов и экспоненциальной функции от числа сжимаемых рядов. Разбиение должно быть проведено таким образом, чтобы процедура свертки и компрессор для ее реализации соответствовали выбранному критерию качества.

Нами исследованы различные подходы к повышению скорости выполнения операции свертки кодов [12—16] и разработке соответствующих методов с использованием многооперандных сумматоров

на основе аппарата симметрических функций. Так, нами предложена модификация процедуры [7], имеющая вид  $m \rightarrow m' \rightarrow 3 \rightarrow 1$ , которая предполагает свертку разработанными нами операционными элементами типа  $m' \rightarrow 3$  [22] на втором этапе и на заключительном этапе — специального сумматора трех слагаемых с параллельным переносом [16], организованного подобно классическому переносу в бинарном сумматоре [19, 21].

Такая модификация позволяет ускорить процедуру свертки вида  $t \rightarrow 1$ . Действительно, параллельные сумматоры 3-1 и 2-1 имеют одинаковые структуры и потому при построении их схем в базисе И—ИЛИ имеют одинаковую задержку. С другой стороны, при одинаковом 1-м этапе процедуры свертки  $m \rightarrow m'$  2-й этап характеризуется такими коэффициентами свертки: при традиционной процедуре  $\kappa_{\text{тр}} = m'/2$ , при модифицированной —  $\kappa_{\text{мод}} = m'/3$ , т. е. на свертку  $m' \rightarrow 3$  требуется меньше времени, чем на  $m' \rightarrow 2$ .

Таким образом, время свертки по модифицированной процедуре меньше времени свертки по традиционной процедуре.

Операция свертки имеет (рис. 3, а) две составляющие: суммирование с формированием внутренних переносов и их преобразование в выходные переносы. Собственно для суммирования используется простая операция сложения по модулю 2. Преобразователь переносов нужен для пересчета значений битов внутренних переносов в значения битов выходных переносов [15] (как правило, они являются позиционными); пересчет необходим в двух случаях: внутренние переносы — непозиционные, либо из старшего разряда имеется несколько внутренних переносов одного веса.

На рис. 3, б и в приведены два вида одnorазрядных операционных элементов — ОЭ1 и ОЭ2; для обоих ОЭ используются одинаковые обозначения: In — биты слагаемых; Sum — бит суммы; Cout — биты внутренних переносов; Cin — биты входных переносов в данный элемент; Carry — биты выходных переносов из данного разряда (для ОЭ2 используется два переноса — в следующий разряд и через разряд).

ОЭ1 представляют собой элементы Уоллеса и Санторо; здесь используются те же обозначения, что и на рис. 2.

К рассмотренным выше переносам можно добавить исследованные нами переносы, явные и неявные. В [8] перенос явно не определяется. В работах [12, 13, 16] рассмотрены многобитовый непозиционный перенос (обобщение переноса Глассера) и позиционный параллельный перенос в различных регулярных фрагментах области битов; в некоторых случаях используется неявный перенос.

В [12—16] предложены методы свертки МРК, использующие симметрию битов каждого РС. В этих методах существенно используется свойство независимости результата от любой перестановки суммируемых битов, из чего и следует их симметрия [17]. В частности, нами предложен табличный метод описания функционирования арифметических устройств

[12]; таблица функционирования по существу является обобщенной таблицей истинности.

Разработанная нами программа позволяет вычислять значения сумм для группы данных без явного вычисления значений внутренних переносов (непосредственное использование свойств симметричных функций, описывающих функционирование сумматора). Именно эти значения и составляют таблицу функционирования; здесь нет необходимости в явном переносе. Схема, построенная по таблице функционирования, обладает наименьшей задержкой из всех схем, рассмотренных выше. Недостатком такого подхода является экспоненциальный рост сложности схемы при росте числа обрабатываемых рядов арифметических кодов.

### Заключение

В обобщенной нами модели предусмотрены различные процедуры и методы обработки области битов произвольной конфигурации, различные операционные элементы для свертки фрагментов многорядных арифметических кодов, выделены внутренние и выходные переносы; модель учитывает организацию быстрых переносов. К настоящему времени разработаны мощные методы суммирования и умножения на его основе при реализации вычислительных операций над группой данных в едином операционном цикле. Такое разнообразие подходов и средств позволяет построить множество альтернативных решений, удовлетворяющих различным время-сложностным критериям.

При малых значениях количества рядов  $m$  ( $m \leq 8$ ) лучшие результаты дает наш подход. В [22] нами предложены ОЭ, позволяющие более быстро свертывать МРК.

Предложенная модель апробирована на многих приложениях. Так, при построении быстродействующего умножителя  $8 \times 8$  [14] проведено разбиение исходного ромба битов частичных произведений на одинаковые подромбы; для них построено несколько вариантов сумматоров типа «ромб». Окончательная свертка осуществляется компрессором 3-1 [16]. Показано, что непосредственное использование аппарата симметрических функций для построения сумматора типа «ромб» в составе умножителя  $8 \times 8$  дает выигрыш в быстродействии в 1,5 раза по сравнению с известным решением, основанным на деревьях Санторо.

### ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Айфичер Э., Джервис Б. Цифровая обработка сигналов.— М.: Изд. дом «Вильямс», 2004.
2. Гамаюн В. П. О развитии многооперандных вычислительных структур // Управляющие системы и машины.— 1990.— № 4.— С. 31—33.
3. Храпченко В. М. Об одном способе преобразования многорядного кода в одnorядный // Докл. АН СССР.— 1963.— Вып. 148.— № 2.— С. 296—299.
4. Храпченко В. М. Методы ускорения арифметических операций, основанные на преобразовании многорядного кода // Вопросы радиоэлектроники. Сер. Электронная вычислительная техника.— 1965.— Вып. 8.— С. 121—144.
5. Брик В. А., Лушпин Л. И. Преобразование многорядного кода в двухрядный при помощи однотипных узлов // Там же.— 1973.— Вып. 7.— С. 94—116.



6. Ромм Я. Е. Метод вертикальной обработки потока целочисленных групповых данных. 1. Групповые арифметические операции // Кибернетика и системный анализ.— 1998.— № 3.— С. 123—151.
7. Гамаюн В. П. Способ ускоренного преобразования многорядного кода в однорядный // Управляющие системы и машины.— 1995.— № 4/5.— С. 10—14.
8. Дрозд А. В., Паулин О. Н., Дрозд Ю. В. Выполнение операции вертикального сложения в арифметических устройствах // Тр. Одес. политехнич. ун-та.— 1997.— Вып. 2.— С. 30—32.
9. Walles C. S. A suggestion for a fast multiplier // IEEE Trans. Comput.— 1964.— Vol. EC-13.— N 1.— P. 14—17.
10. Santoro M. R. Design and clocking of VLSI multipliers / Stanford University, Computer Systems Laboratory. Report Number: CSL-TR-89-397, October 1989.
11. Паулин О. Н., Ляховецкий А. М. Модифицированный метод суммирования многорядных кодов на основе многооперандных сумматоров // Тр. Одес. политехнич. ун-та.— 1999.— Вып. 3.— С. 147—148.
12. Паулин О. Н., Ляховецкий А. М. Модель и метод проектирования многооперандных сумматоров на базе симметрических функций // Тези доповідей на міжнар. конф. з індуктивного моделювання МКІМ-2002.— Львів.— 2002.— С. 208—213.
13. Паулин О. Н. К построению быстродействующих арифметических устройств / В сб.: Искусственный интеллект. Спецвыпуск. Т. 3.— Донецьк: ІПШІ, 2002.— С. 314—322.
14. Паулин О. Н. К разработке умножителя на основе сумматора типа ромб / В сб.: Искусственный интеллект. Т. 4.— Донецьк: ІПШІ, 2006.— С. 35—41.
15. Paulin O. N. On carry organization in compressing multi-row codes / Pattern Recognition and Information Processing (PRIP'05): Proceedings of the Eighth International Conference.— Minsk.— 2005.— P. 454—456.
16. Паулин О. Н. О свертке трехрядных кодов // Управляющие системы и машины.— 2005.— № 5.— С. 68—72.
17. Паулин О. Н. К построению прикладной теории симметрических булевых функций / В сб.: Искусственный интеллект. Т. 4.— Донецьк: ІПШІ, 2005.— С. 245—255.
18. Ромм Я. Е. Метод вертикальной обработки потока целочисленных групповых данных. II. Приложение к бинарным арифметическим операциям // Кибернетика и системный анализ.— 1998.— № 6.— С. 146—162.
19. Угрюмов Е. П. Цифровая схемотехника.— СПб: БХВ-Петербург, 2004.
20. Pat. 3906211 USA. Three-word adder carry propagation / A. V. Glasser; Bell Telephone Lab.— 1975.
21. Карцев М. А., Брик В. А. Вычислительные системы и синхронная арифметика.— М.: Радио и связь, 1981.
22. Нестеренко С. А., Паулин О. Н. К синтезу операционных элементов типа m-3 // Мат-лы МНПК «Информационные технологии и информационная безопасность в науке, технике и образовании «Инфотех-2007». Ч. 1.— Севастополь.— 2007.— С. 40—43.

НОВЫЕ КНИГИ

**Новожилов О. П. Основы микропроцессорной техники. В 2-х томах. Том 1.— М.: РадиоСофт, 2007.— 432 с.**

Излагаются основные концепции микропроцессорной техники, структурно-функциональные особенности микропроцессоров, микроконтроллеров и сигнальных процессоров, современные технологии обработки цифровой информации и средства их обеспечения. Книга состоит из двух томов, в которые вошли три раздела, и написана как учебное пособие.

Первый раздел служит введением в микропроцессорную технику: рассмотрены общие принципы построения процессоров; приведены основные понятия и начальные сведения о микропроцессорах и микропроцессорных системах; рассмотрены структурно-функциональная организация процессоров, регистровые модели, способы адресации, система команд и организация циклов их выполнения. Излагаются вопросы структурно-функциональной организации микропроцессорных систем: рассмотрены их типовые структуры и аппаратные средства, вопросы организации ввода/вывода, прерываний и прямого доступа к памяти.

Во втором разделе излагаются общие вопросы структурно-функциональной организации микроконтроллеров, а также средства их программирования и отладки. Приводится материал по конкретным контроллерам с RISC- и CISC-архитектурой. Рассмотрены структурно-функциональная организация, программные модели и система команд сигнальных процессоров, предназначенных для вычислений с фиксированной точкой.

В третьем разделе рассмотрены архитектурные и другие особенности универсальных процессоров, вопросы взаимодействия процессора с основной памятью (концепция и средства взаимодействия, организация основной памяти и кэша, защита памяти и многозадачность); процессорные технологии (обработка прерываний и исключений, представление и обработка данных с плавающей точкой, технологии MMX и SSE, суперскалярная и мультискалярная технологии). Приведен материал по конкретным семействам современных универсальных процессоров.

В первый том вошли 1-й и 2-й разделы.

Книга адресуетеся всем желающим ознакомиться с основами микропроцессорной техники и, в первую очередь, студентам ВУЗов и техникумов.

