

Т.Б. Мартынюк, А.В. Кожемяко, В.В. Хомюк

## Модели систолических массивов для обработки векторных данных по разностным срезам

Показана последовательность и результаты синтеза систолического процессора методом С. Куна. Учтены особенности обработки векторных данных по разностным срезам.

A sequence and the results of the synthesis of a systolic processor by the method of canonic images of basis regular iterative algorithms on a systolic array by S. Kung's method are shown. The features of the vector data processing by difference slices are taken into account.

Показано послідовність і результати синтезу систолического процесора за методом С. Куна. Враховано особливості оброблення векторних даних за різницевиими зрізами.

**Введение.** В создании суперкомпьютеров приоритетными считаются разработки специализированных процессоров с оптимальным сочетанием свойств векторных, мультиматричных процессоров и процессоров с длинным командным словом [1]. В свою очередь, известно, что систолические процессоры (СП) представляют собой специализированные мультиматричные устройства, конфигурируемые в соответствии с алгоритмом для решения конкретной задачи. При этом практика показала, что для эффективной реализации широкого класса алгоритмов обработки сигналов и изображений [2–4] перспективны сетевые вычислительные структуры с локальными связями, т.е. систолические и волновые матрицы. Это объясняется тем, что основную часть вычислений этих алгоритмов можно свести к некоторому базовому множеству операций над матрицами, относящихся к классу рекурсивных локально-зависимых алгоритмов, а для систолических и волновых массивов характерной особенностью являются локальные и регулярные перемещения данных [2–5]. Кроме того, систолические массивы представляют класс конвейерных матричных архитектур и имеют такие свойства, как модульность, регулярность, локальность связности и временную локальность, конвейерность и синхронность мультиобработки данных [4, 6, 7].

Известно, что сложность отображения конкретного алгоритма на систолический массив – это не только необходимость выполнения условия принадлежности этого алгоритма к регулярным итерационным алгоритмам (РИА) [7–8], но и сложность проведения оптимизации на множестве возможных систолических архитектур [9–10]. И только благодаря разработке формальных методов отображения РИА на систолические матрицы [2, 9–13] были получены систолические версии для большинства алгоритмов обработки сигналов и изображений, линейной алгебры, задач математической физики и т.д.

### Постановка задачи

В общем случае аппаратный синтез систолических матриц на базе графов содержит следующие этапы [14]:

- запись алгоритмов в виде, пригодном для синтеза, т.е. как набор функциональных шагов;
- отображение набора функциональных шагов в пространственно-временные координаты;
- проектирование систолических матриц из пространственно-временного графа.

Первый этап сводится к представлению алгоритма либо в виде графа зависимостей (ГЗ) [8], либо решетчатого функционального графа (РФГ) [10] с использованием пространственно-временного индексирования и соответствующих дуг для отображения зависимостей в индексном пространстве [2].

На втором этапе ГЗ отображается на массив графа потока сигналов (ГПС) с привлечением операции назначения на процессоры и опера-

---

**Ключевые слова:** систолический процессор, регулярный итерационный алгоритм, разностный срез, граф зависимостей, граф потока сигналов.

ции планирования [2], или РФГ отображается посредством пространственно-временного оператора на оптимизированную структурную схему систолического массива [9, 10].

Получение функции синхронизации, т.е. многомерной развертки во времени выполнения функциональных операторов РФГ алгоритма [11], можно отнести к третьему этапу. Для превращения ГПС в систолический массив на третьем этапе используется процедура систолизации (ресинхронизации), базирующаяся на соответствующих сечениях [2, 4].

Наиболее известны и детально представлены в литературе два метода отображения на систолические структуры:

- метод канонического отображения, предложенный С. Куном [2];
- формализованный метод проектирования параллельных вычислительных структур [13].

Для первого метода характерна более простая и наглядная форма алгебраических преобразований, для второго разработаны прикладные программы проектирования СП [15]. В данной работе, учитывая простоту исходного алгоритма обработки векторных массивов, используется первый метод.

Один из наиболее распространенных в алгоритмах обработки сигналов и изображений – оператор группового суммирования (ГС) вида [16]

$$X_n = \sum_{i=1}^n a_i, \quad (1)$$

где  $a_i$  – элементы исходного массива данных.

Если ограничиться рассмотрением векторного массива, то операция (1) известна как свертка элементов векторного массива [17], которая, будучи интегральной операцией, трудно поддается распараллеливанию [18]. Среди известных параллельных алгоритмов, реализующих оператор ГС, наибольшее распространение получил алгоритм логарифмического суммирования или рекурсивного сдвигания [19], выполняющийся по древовидной (пирамидальной) схеме сумматоров за время  $O(\log_2 n)$ . Альтернативный параллельный алгоритм – это ал-

горитм, в котором используется обработка векторных массивов по разностным срезам (РС) [20]. Он может быть классифицирован также, как алгоритм многооперандного суммирования.

Суть обработки по РС заключается в том, что в каждом  $j$ -м цикле обработки ( $j = \overline{1, N}$ ) все элементы векторного массива  $A_{j-1}$  уменьшаются на конкретную величину  $q_j$ , которую можно рассматривать как внутренний порог обработки, значение которого совпадает с минимальным ненулевым элементом текущего массива  $A_{j-1}$  [21]. Последовательность шагов в каждом  $j$ -м цикле алгоритма можно представить следующим образом.

Шаг 1. Определяется общая значимая часть всех элементов массива  $A_{j-1}$  в  $j$ -м цикле, т.е.  $j$ -й внутренний порог  $q_j$  обработки вида

$$q_j = \min A_{j-1} = \min \{a_{i,j-1}\}_{i=1}^n, \quad j = \overline{1, N}, \quad (2)$$

где  $A_0$  – начальный векторный массив чисел.

Проверяется условие:

$$q_j = 0. \quad (3)$$

Если оно выполняется, то процесс суммирования заканчивается, если – нет, то выполняется переход к шагу 2.

Шаг 2. Формируется разностный срез (РС)  $A_j$  вида

$$A_j = \{a_{i,j}\}_{i=1}^n = \{a_{i,j-1} - q_j\}_{i=1}^n. \quad (4)$$

В дальнейшем полученный РС  $A_j$  представляет собой входной массив слагаемых для следующего  $(j+1)$ -го цикла. Формируется бинарная маска  $f_j$  с элементами  $f_{i,j}$  вида:

$$f_{i,j} = \begin{cases} 1, & \text{если } a_{i,j} \geq 0, \\ 0, & \text{если } a_{i,j} < 0. \end{cases} \quad (5)$$

Шаг 3. Формируется частичная сумма  $S_j$  вида

$$S_j = q_j p_j = q_j \cdot \sum_{i=1}^n f_{i,j}, \quad (6)$$

где  $p_j$  – количество положительных ненулевых элементов массива  $A_{j-1}$ .



следний элемент ( $i = n$ ) которой является минимальным элементом  $\{a(i)\}$ . При этом  $\alpha(0) = \infty$ . Рекурсивные уравнения и принятые обозначения величин для данного базового алгоритма приведены в таблице. Рекурсивное уравнение для величины  $a_i^j$  представлено в общем виде, поскольку функция  $\psi$  будет определена позже. На рис. 2,а представлен ГЗ базового алгоритма выборки величины  $q_j$ , на рис. 2,б показан узел ГЗ с назначенными операциями. Входная последовательность  $a(i)$  поступает на вход верхней строки ГЗ как  $a_{i,0}$ , данные  $\alpha_{0,j}$  устанавливаются в узлах первого столбца ГЗ значения  $\infty$ . Последний  $n$ -й элемент последовательности  $\alpha(i)$  появляется на выходе правого столбца ГЗ в виде  $\alpha_{n,j} = q(j)$ .

Анализ ГЗ на рис. 2,а позволяет определить  $\alpha_i^j$  как передаваемые данные, а ГЗ – как инвариантный относительно сдвига, поскольку структура зависимостей от узла к узлу не изменяет-

ся. Свойство ассоциативности процедуры выборки минимальной величины позволяет использовать реверсивные дуги в ГЗ, соответствующие ассоциативным операциям, показанным на рис. 2,б утолщенной линией.

**Алгоритм формирования разностных срезов  $A_j$ .** Рекурсивные уравнения и принятые обозначения переменных для этого базового алгоритма приведены в таблице, а на рис. 3,а показан соответствующий ГЗ с локальными зависимостями данных, на рис. 3,б – детализация узла ГЗ. На рис. 3,а используется обозначение для величин  $a_i^j = r(i, j)$ ,  $j = \overline{1, n}$ , где  $r(i, j)$  –  $i$ -й элемент РС  $A_j$  в  $j$ -й рекурсии.

Анализ рис. 2 и 3 позволяет сделать вывод, что существует возможность совместного выполнения двух рассмотренных выше базовых алгоритмов, поскольку функция  $\psi$  для алгоритма выборки может быть представлена в виде операции вычитания  $a_i^j = \psi(a_i^{j-1}, q_j) =$

$= a_i^{j-1} - q_j$  в соответствии с выражением (4).

Кроме того, необходимо отметить, что данные  $q(j)$  могут быть переданы в обоих направлениях, что показано реверсивной дугой на рис. 3,б. В каждом узле на рис. 3,б необходимо также сформировать соответствующий элемент  $f(i, j)$  бинарной маски  $f_j$  вида (5).

**Алгоритм формирования суммы  $S_N$  массива чисел.** Рекурсивные уравнения и принятые обозначения величин для данного базового алгоритма приведены в таблице (1-й вариант), соответствующий ГЗ представлен на рис. 4,а, а детализация узлов ГЗ – на

Т а б л и ц а

Составляющие канонического отображения на матричные структуры	Алгоритм выборки минимальной величины в массиве чисел	Алгоритм формирования разностных срезов	Алгоритм формирования (накопления) суммы массива чисел
Аналитические зависимости	$q_j = \min_i A_{j-1} = \min_i \{a_{i,j-1}\}_{i=1}^n$	$A_j = \{a_{i,j}\}_{i=1}^n = \{a_{i,j-1} - q_j\}_{i=1}^n$	$S_j = S_{j-1} + \sum_{i=1}^n f_{i,j} \cdot q_j$
Рекурсивные уравнения	$\alpha_{i+1}^j = \min_i (a_i^{j-1}, \alpha_i^j)$ $a_{i+1}^j = \psi(a_i^{j-1}, q_i^j)$ $\alpha_0^j = \infty$ $q(j) = \alpha_n^j$	$a_i^j = a_i^{j-1} - q_j^j$ $q_{i+1}^j = q_i^j$	1-й вариант: $S_i^j = S_{i-1}^j + q_i^j \cdot f_i^j$ $q_{i+1}^j = q_i^j$ $S_0^1 = 0; S_0^j = S_n^{j-1}$ 2-й вариант: $S_i^j = S_{i-1}^j + q_i^j \cdot f_i^j$ для $\forall i$ , кроме $i = n$ $S_n^j = S_n^{j-1} + S_{n-1}^j + q_n^j \cdot f_n^j$ для $i = n$ $q_{i+1}^j = q_i^j$ $S_0^j = 0$ 3-й вариант: $S_i^{2j} = S_{i+1}^{2j} + q_i^j \cdot f_i^j$ $S_i^{2j-1} = S_{i-1}^{2j-1} + q_i^j \cdot f_i^j$ $q_{i+1}^{2j-1} = q_i^{2j-1}; q_i^{2j} = q_{i+1}^{2j}$ $S_0^1 = 0; S_0^{2j+1} = S_0^{2j}; S_0^{2j} = S_n^{2j-1}$
Принятые обозначения переменных	$a_i^j = a(i, j) = a_{i,j}$ $\alpha_i^j = \alpha(i, j) = \alpha_{i,j}$	$a_i^j = a(i, j) = a_{i,j}$ $q_i^j = q(j)$ , так как $q_j^j = const$ для $\forall_i$	$S_i^j = S(i, j) = S_{i,j}$ $q_i^j = q(j)$ , так как $q_j = const$ для $\forall_i$ $f_i^j = f(i, j) = f_{i,j}$

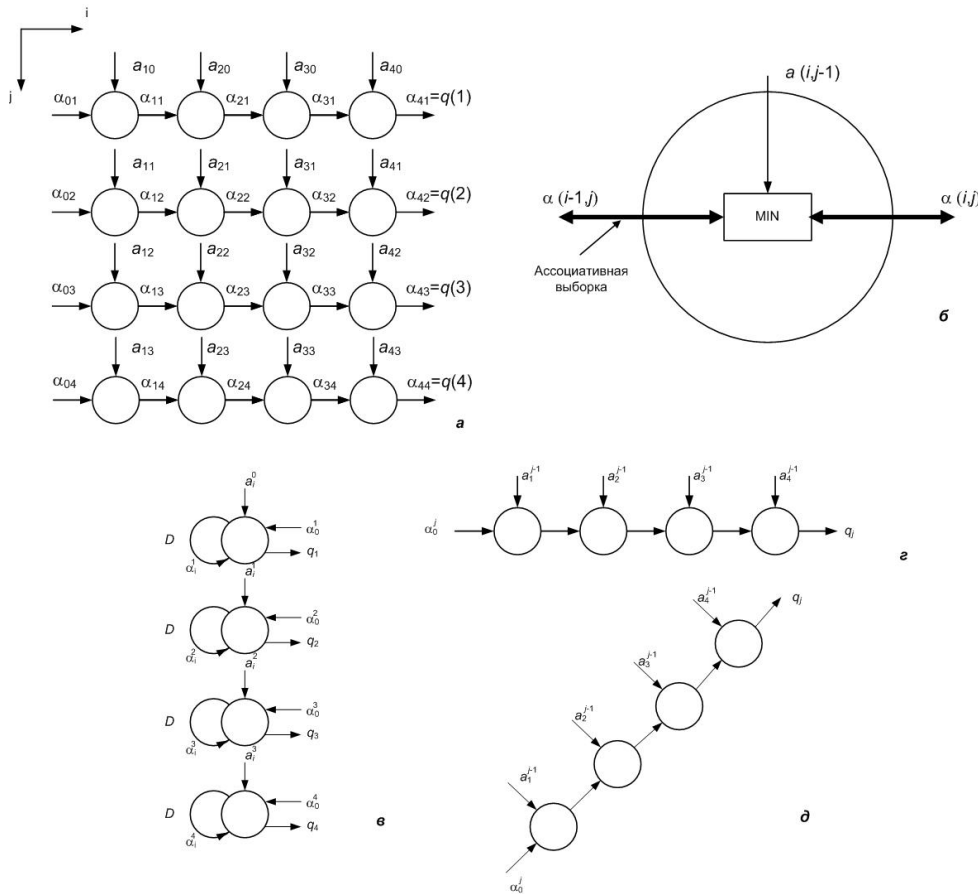


Рис. 2

рис. 4,б. Анализ приведенного ГЗ (рис. 4,а) показывает, во-первых, что данный ГЗ совместим с рассмотренными ранее ГЗ для базовых алгоритмов выборки  $q_j$  и формирования РС, а во-вторых, что при выполнении данного алгоритма необходимо наличие соответствующих элементов  $f_{i,j}$  в каждом узле ГЗ (рис. 4,б), что также может обеспечить базовый алгоритм формирования РС (рис. 3,б).

В отношении ГЗ на рис. 4,а необходимо отметить, что он является спиральным ГЗ, поскольку для передачи результата из узла  $(n, j)$  в узел  $(1, j+1)$  необходима спиральная (не локальная) связь (1-й вариант). Локализация связи для промежуточной переменной  $S_n^j$  может быть выполнена двумя способами. Для одного из них характерны рекурсивные уравнения, представленные в таблице как 2-й вариант, а соответствующий локализованный ГЗ показан на рис. 4,в. Другой способ реализован с исполь-

зованием свойства ассоциативности операции суммирования, что позволяет записать рекурсивный алгоритм в виде 3-го варианта, представленного в таблице. Его особенность – сохранение справедливости рекурсивных уравнений 1-го варианта таблицы для переменных с временными нечетными индексами  $(2j-1)$ . Соответствующий локализованный ГЗ представлен на рис. 4,з, причем два возможных направления потока данных  $S(j)$  и  $q(j)$  показаны на рис. 4,б утолщенной линией с двумя стрелками.

Процесс отображения ГЗ в ГПС можно осуществить за два шага [2]. На первом шаге выполняется назначение на процессорные элементы (ПЭ) с использованием линейной проекции, когда ГЗ отображается на решетку меньшей размерности, представляющую собой процессорное пространство. При линейном отображении все узлы ГЗ, расположенные на одной прямой линии, отображаются в один ПЭ. Математически линейная проекция представляется вектором  $\vec{a}$  [2]. На втором шаге выполняется планирование работы ПЭ, когда многомерное индексное пространство ГЗ отображается на одномерное пространство планов (расписаний). Например, линейный план отображает множество параллельных одновременных гиперплоскостей в множество линейно-возрастающих временных индексов. Математически план представляется вектор-столбцом плана  $\vec{S}$ , определяющим направление, перпендикулярное к гиперплоскостям в ГЗ [2].

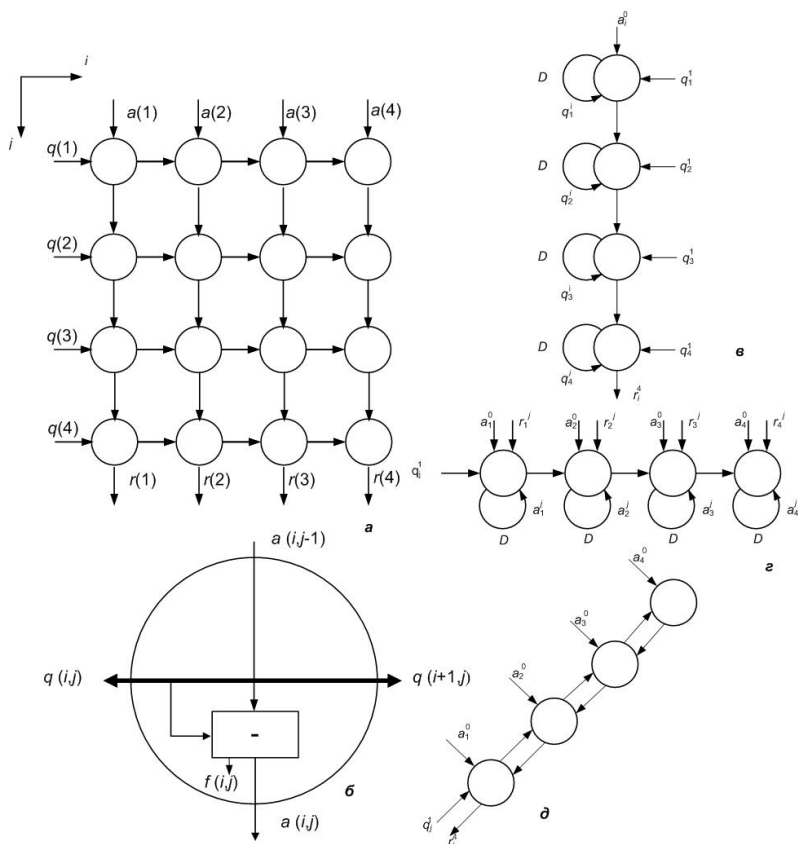


Рис. 3

Из ГЗ для базового алгоритма выборки минимальной величины в массиве числовых величин (рис. 2,а) путем проекции в определенном направлении можно получить соответствующий ГПС.

■ *Проекция в направлении* [10], когда  $\vec{d}^T = \vec{S}^T = [10]$ . Итоговый ГПС (рис. 2,в) представляет собой вертикальный линейный массив. Входные данные  $a_i^{j-1}$  ( $j-1$ -го массива  $A_{j-1}$ ) последовательно поступают в  $j$ -й узел, где формируются исходные данные  $q_j$ , которые выводятся параллельно по шине вывода. В каждом узле есть вход для  $\alpha_0^j = \infty$ .

■ *Проекция в направлении* [01] или в направлении оси  $j$ , когда  $\vec{d}^T = \vec{S}^T = [01]$ . Итоговый ГПС в этом случае представляет собой горизонтальный линейный массив (рис. 2,г), в котором входные данные  $a_i^{j-1}$  ( $j-1$ -го массива  $A_{j-1}$ ) параллельно поступают во все  $n$  узлы, пе-

ременные проходят последовательно слева направо, результат  $q_j$  выводится последовательно.

■ *Проекция в направлении* [11], когда  $\vec{d}^T = \vec{S}^T = [11]$ . Итоговый ГПС также представляет собой линейный массив (рис. 2,д), в котором переменные проходят последовательно слева направо, результат  $q_j$  выводится последовательно, а входные данные  $a_i^{j-1}$  ( $j-1$ -го массива  $A_{j-1}$ ) вводятся параллельно в  $i$ -е узлы, но последовательно в  $i$ -м такте  $j$ -го цикла работы.

Аналогично из ГЗ для алгоритма формирования РС (рис. 3а) путем проекций в трех направлениях можно получить три вида ГПС.

■ *Проекция в направлении* [10]. Итоговый ГПС (рис. 3,в) представляет собой вертикальный линейный массив, в котором входные данные  $a_i^0$  массива  $A_0$  последовательно поступают в первый узел, а исходные данные  $r_i^n$  формируются в  $n$ -м узле и выводятся последовательно.

■ *Проекция в направлении* [01]. Итоговый ГПС (рис. 3,г) представляет собой горизонтальный линейный массив, в котором входные данные  $a_i^0$  массива  $A_0$  параллельно поступают во все  $n$  узлы, переменные  $q_j$  проходят последовательно слева направо через все узлы, а результат  $r_i^j$  формируется в соответствующем  $i$ -м узле и может выводиться параллельно в  $j$ -м цикле

■ *Проекция в направлении* [11]. Итоговый ГПС (рис. 3,д) представляет собой линейный массив, в котором переменные  $q_j$  и  $r_i^j$  передвигаются в противоположных направлениях, а входные данные  $a_i^0$  массива  $A_0$  поступают в соответствующий  $i$ -й узел, но последовательно в  $i$ -м такте нулевого цикла.

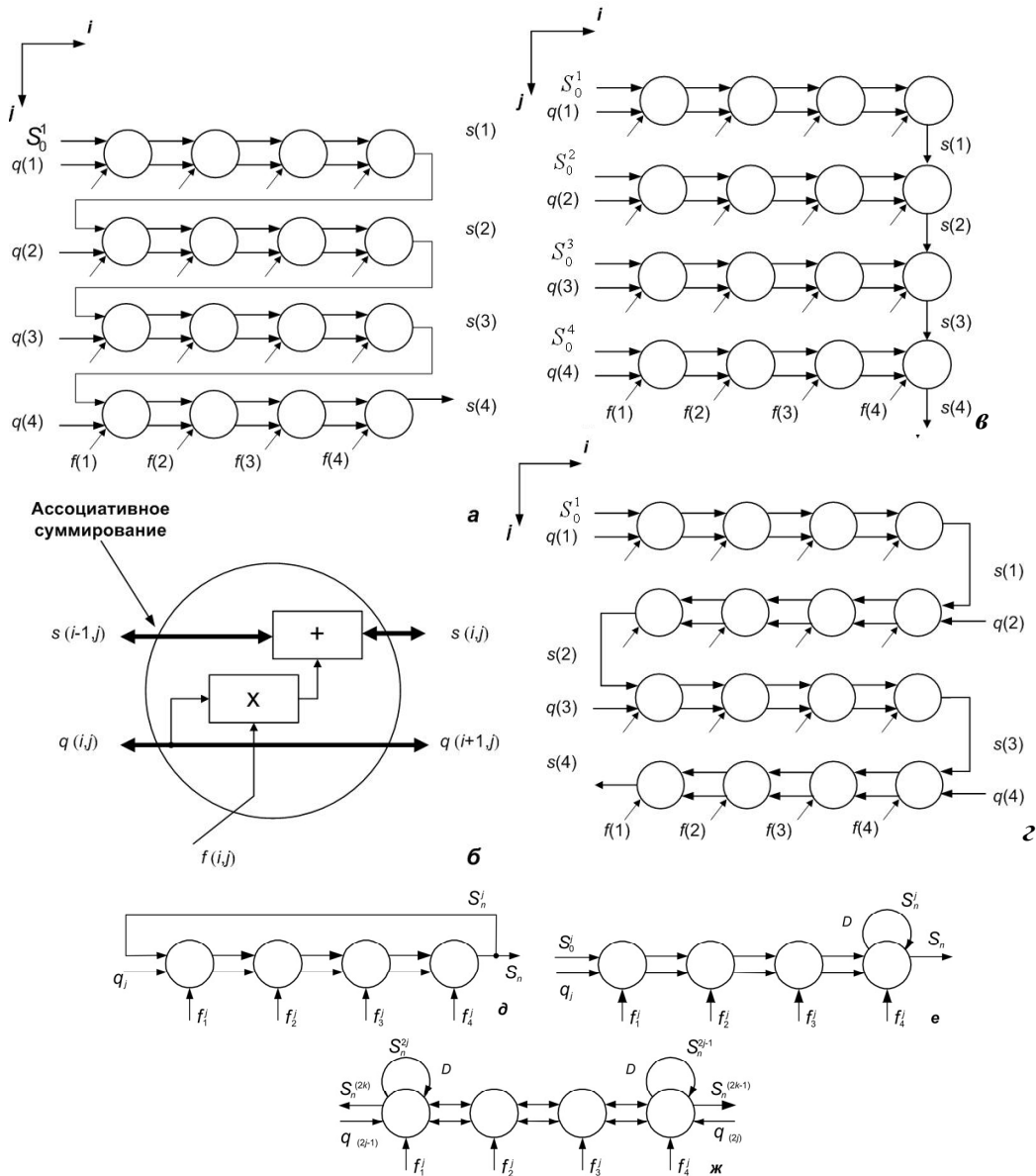


Рис. 4

Рассмотрим реальность выполнения полученных ГПС для данного базового алгоритма. Поскольку выполнение базового алгоритма выборки минимальной величины в массиве числовых величин (рис. 2,а) диктует последовательное формирование величины  $q_j$ , то не имеет смысла использовать ГПС, в котором предусмотрен параллельный вывод величин  $q_j$  (рис. 3,в). Кроме того, поскольку при параллельном вводе входных данных  $a_i^j$  желательно получить в параллельной форме результат  $r_i^j$ , который является входной величиной  $a_i^j$  в

следующем  $(j+1)$ -м цикле работы, то нет необходимости использовать ГПС с последовательным выводом результата (рис. 3,д). Таким образом, для последующего рассмотрения остается проекция в ГПС (рис. 3,з) в направлении [01].

Учитывая наличие трех вариантов ГЗ для алгоритма накопления суммы  $S_j$  массива числовых величин (рис. 4, а, в, з), рассмотрим проекцию в ГПС в выбранном направлении [01] для всех трех вариантов.

1. Итоговый ГПС для ГЗ (рис. 4,а) представляет собой горизонтальный линейный массив (рис. 4,д), в котором переменные  $q_j$ ,  $S_j$  проходят последовательно слева направо, переменные  $f_i^j$  поступают параллельно во все  $n$  узлы, результат  $S_n$  формируется в  $n$ -м цикле в  $n$ -м узле. Кроме того, в ГПС существует петля между  $n$ -м и первым узлом для передачи величины  $S_n^j$ , соответствующая глобальной связи в ГЗ (рис. 4,а).

2. Итоговый ГПС для ГЗ (рис. 4,в) представляет собой горизонтальный линейный массив (рис. 4,е), в котором переменные  $q_j$  и  $S_j$  двигаются слева направо, переменные  $f_i^j$  посту-

тупают параллельно во все  $n$  узлы, результат  $S_n$  формируется в  $n$ -м цикле в  $n$ -м узле. Кроме того, в ГПС существует петля между  $n$ -м и первым узлом для передачи величины  $S_n^j$ , соответствующая глобальной связи в ГЗ (рис. 4,а).

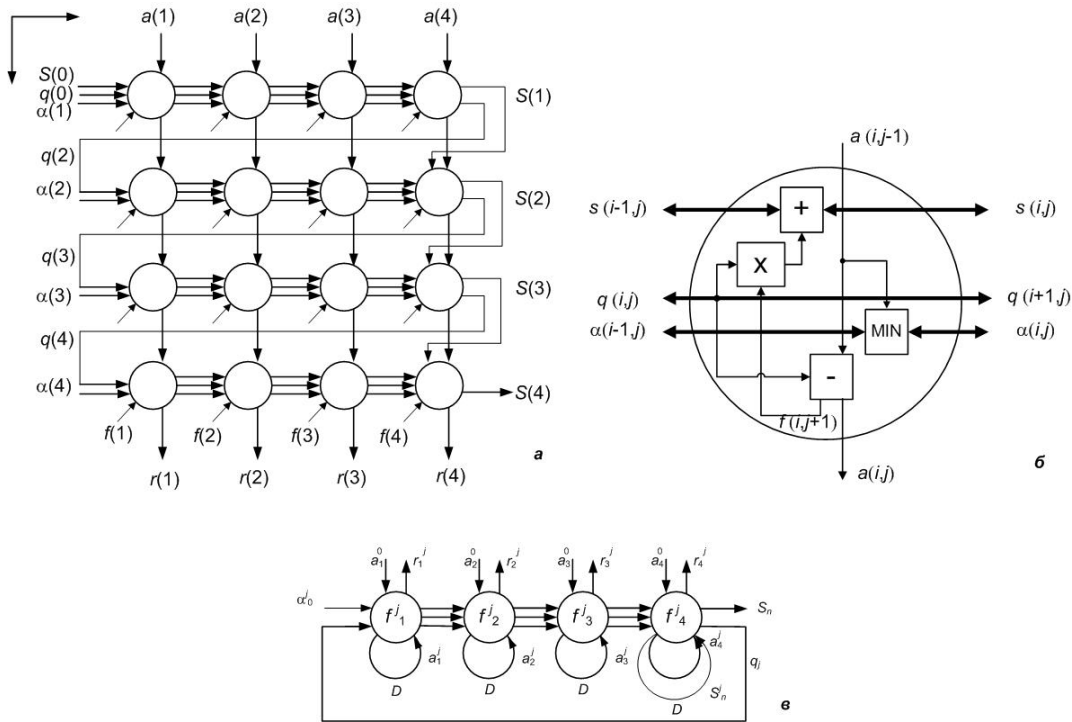


Рис. 5

пают параллельно во все  $n$  узлы, результат  $S_n$  формируется в  $n$ -м цикле в  $n$ -м узле и присутствует петля ( $S_n^j$ ) у  $n$ -го узла, соответствующая локальной связи в ГЗ (рис. 4,б).

3. Итоговый ГПС для ГЗ (рис. 4,з) представляет собой горизонтальный линейный массив (рис. 4,жс), в котором в нечетные  $(2j-1)$ -е циклы переменные  $q_{(2j-1)}, S_{(2j-1)}$  проходят слева направо, а в четные  $(2j)$ -е циклы переменные  $q_{(2j)}, S_{(2j)}$  двигаются наоборот. Соответственно, результат  $S_n^{(2k-1)}$  формируется в  $n$ -м узле при  $n$  – нечетном, а результат  $S_n^{(2k)}$  – в первом узле при  $n$  – четном ( $k = 1, 2, \dots$ ). Переменные  $f_i^j$  поступают параллельно во все  $n$  узлы массива. В ГПС есть две петли  $S_n^{(2j)}$  и  $S_n^{(2j-1)}$  у первого и  $n$ -го узлов, они соответствуют локальным связям в ГЗ (рис. 4,з).

Для объединенного алгоритма многооперандного суммирования по РС целесообразно выбрать ГЗ (рис. 4,б) и соответствующий ГПС (рис. 4,е) для базового алгоритма формирования и накопления частичных сумм.

Такой выбор объясняется тем, что этот вариант занимает промежуточное положение между 1-м и 3-м вариантом, поскольку имеет только одну глобальную связь для переменной  $q_j$  в отличие от 1-го варианта. Вместе с тем, именно для этой переменной возникают сложности при аппаратной реализации реверсивных дуг, что характерно для 3-го варианта.

На рис. 5,а показан ГЗ для объединенного алгоритма, на рис. 5,б – детализация узла ГЗ, а на рис 5,в – соответствующий ГПС. Итоговый ГПС представляет собой горизонтальный линейный массив, в котором переменные  $\alpha_j, q_j, S_j$  проходят последовательно слева направо, переменные  $a_i^0$  поступают параллельно во все  $n$  узлы массива, результат  $S_n$  формируется в  $n$ -м узле в  $n$ -м цикле, в  $j$ -м цикле параллельно во всех  $n$  узлах формируется результат  $r_i^j$ . В каждом  $i$ -м узле массива формируются переменные  $f_i^j$  и существует петля ( $a_i^j$ ). Кроме того, в ГПС присутствует вторая петля ( $S_n^j$ ) в  $n$ -м узле, соответствующая локальным связям в ГЗ (рис. 5,а).

Результатом проектирования СП с учетом ГПС (рис. 5,в) на третьем этапе синтеза является систолический процессор, структурно представленный на рис.6а, с соответствующей функциональной организацией ПЭ (рис. 6,б). Основными узлами ПЭ являются: комбинационные сумматоры  $SM$  и  $SM^*$  (в режиме вычита-



ния), три регистра  $RGR$ ,  $RGN$ ,  $RGM$ , миниматор  $MIN$ , мультиплексор  $MX$ ,  $D$ -триггер и блок логических элементов И (&) [23]. Старший ПЭ отличается от остальных тем, что сумматор  $SM$  у него накапливающий. Каждый ПЭ имеет пять информационных входов: вход операнда  $a_{i,0}$  и четыре входа для связи с предыдущим ПЭ ( $S_{i-1,j}$ ,  $q_{i-1,j}$ ,  $q_{i-1,j-1}$ ,  $Sign q_{i-1,j}$ ), а также четыре информационных выхода (кроме старшего ПЭ) для связи с последующим ПЭ ( $S_{i,j}$ ,  $q_{i,j}$ ,  $q_{i,j-1}$ ,  $Sign q_{i,j}$ ) [24]. Управляющие входы и входы синхронизации на рис. 6 не показаны.

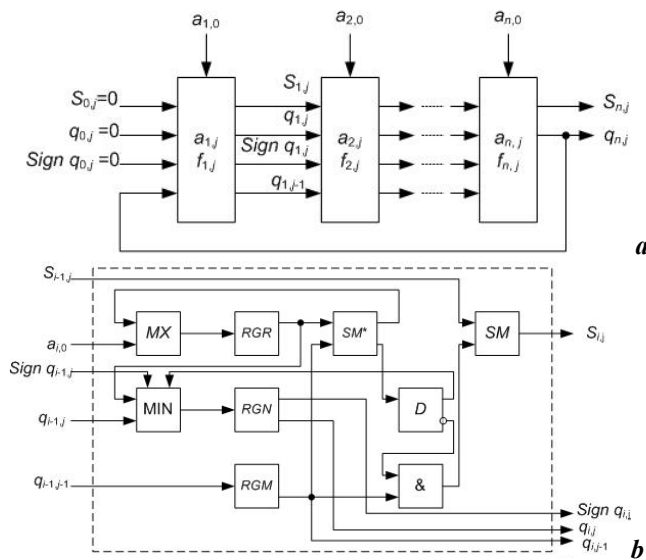


Рис. 6

В рабочем режиме ПЭ выполняет следующие операции:

- выделение текущего минимального значения  $q_{i,j}$  двух операндов ( $a_{i,j}$ ,  $a_{i-1,j}$ ) и его знака  $Sign q_{i,j}$  (задействованы миниматор  $MIN$  и регистр  $RGN$ );
- формирование элемента  $a_{i,j}$  текущего разностного среза и элемента  $f_{i,j}$  (задействованы сумматор  $SM^*$ ,  $D$ -триггер, регистры  $RGR$ ,  $RGM$ );
- формирование текущей суммы  $S_{i,j}$  (задействованы сумматор  $SM$ , регистр  $RGM$  и блок &).

Конвейерная организация вычислительного процесса в СП позволяет обеспечить его ритмичную работу без простоя аппаратных узлов, о чем свидетельствует анализ временных диа-

грамм работы ПЭ [25]. Регулярность структуры СП обеспечивает эффективное «погружение» реализуемого алгоритма многооперандного суммирования массива чисел по РС в среду ПЛИС. Результаты «размещения» предложенной структуры СП на одной микросхеме ПЛИС  $EP1K100FC256-3$  в виде многовходового сумматора показали его функционирование в миллисекундном диапазоне [26].

Среди областей эффективного использования такого сумматора можно отметить:

- корреляционную обработку сигналов и изображений [27];
- адаптивные системы управления для промышленных роботов [28];
- распознавание биоэлектрических сигналов [29];
- моделирование пороговых нейронов [21, 30].

**Заключение.** Для отображения базовых регулярных итерационных алгоритмов для многооперандного суммирования известный метод С. Куна адаптирован с учетом особенности обработки векторных массивов по разностным срезам, что позволило формализовать процесс проектирования оптимальной структуры систолического процессора.

Анализ результатов моделирования предложенной структуры систолического процессора на ПЛИС показал, что существует возможность его реализации в виде микросхемы многовходового сумматора с ориентировочным количеством входов от 20 до 60 в зависимости от характеристик конкретных семейств ПЛИС.

Аппаратная реализация арифметическо-логических операций ПЭ систолического процессора на ПЛИС значительно сокращает время обработки, что обеспечивает эффективное использование предложенного многовходового сумматора для обработки сигналов и изображений в системах реального времени, где необходимое время отклика должно составлять миллисекунды и меньше.

1. Митропольский Ю.И. Суперкомпьютеры и микропроцессоры: Приоритеты исследований и разработок

- // Электроника: Наука. Технология. Бизнес. – 2000. – № 2. – С. 18–21.
2. Кун С. Матричные процессоры на СБИС. – М.: Мир, 1991. – 672 с.
  3. Кухарев Г.А., Шмерко А.Ю., Шмерко В.П. Алгоритмы и систолические процессоры для обработки многозначных данных. – Мн.: Наука і тэхніка, 1990. – 296 с.
  4. Гун С. Систолические и волновые матричные процессоры для высокопроизводительных вычислений // ТИИЭР. – 1984. – 72, № 7. – С. 133–53.
  5. Каневский Ю.С., Сергиенко А.М. Локально связанные вычисления – состояние и перспективы. – <http://www.icfst.kiev.ua/symposium/proceedings/kanevsky.doc>
  6. Молдован Д.И. О разработке алгоритмов для систолических матриц СБИС // ТИИЭР. – 1983. – 91, № 1. – С. 140–49.
  7. Джагадиш Х.В., Рао С.К., Кайлат Т. Матричные структуры для реализации итерационных алгоритмов // Там же. – 1987. – 75, № 9. – С. 184–203.
  8. Рао С.К., Кайлат Т. Регулярные итеративные алгоритмы и их реализация в процессорных матрицах // Там же. – 1988. – 76, № 3. – С. 58–69.
  9. Каневский Ю.С., Корчев Д.В. Способ отображения множества индексов переменных алгоритма на систолическую структуру // Кибернетика. – 1991. – № 1. – С. 51–57.
  10. Выжиковски Р., Елфимова Л.Д., Каневский Ю.С. Реализация на систолических массивах некоторых итерационных алгоритмов решения систем линейных алгебраических уравнений // Кибернетика и системный анализ. – 1992. – № 5. – С. 145–158.
  11. Каневский Ю.С., Корчев Д.В. Метод синтеза многомерной временной развертки вычислений в систолических процессорах // Электронное моделирование. – 1990. – 12, № 4. – С. 43–49.
  12. Каневский Ю.С. Систолические процессоры. – К.: Техніка, 1991. – 173 с.
  13. Арифметика, принципы организации, диагностика и формализованное проектирование вычислительных структур и устройств / В.П. Тарасенко, Н.В. Черкасский, Ю.С. Каневский и др. // К.: Вища шк., 1989. – 343 с.
  14. Аксенов В.П., Красинский П.Я., Спиридонов Г.В. Систолические алгоритмы и процессоры // Зарубежная радиоэлектроника. – 1987. – № 7. – С. 7–33.
  15. Каневский Ю.С., Овраменко С.Г., Синичук И.И. Архитектурный компилятор систолических структур // УСиМ. – 1991. – № 3. – С. 10–21.
  16. Справочник по цифровой вычислительной технике / Под ред. В.Н. Малиновского. – К.: Техніка, 1980. – 320 с.
  17. СБИС для распознавания образов и обработки изображений: Пер. с англ. / Под ред. К. Фу. – М.: Мир, 1988. – 248 с.
  18. Карцев М.А., Брик В.А. Вычислительные системы и синхронная арифметика. – М.: Радио и связь, 1981. – 360 с.
  19. Системы параллельной обработки / Под ред. Д. Ивенса. – М.: Мир, 1985. – С. 285–337.
  20. Мартинюк Т.Б. Рекурсивні алгоритми багатооперандної обробки інформації. – Вінниця: «УНІВЕРСУМ-Вінниця», 2000. – 216 с.
  21. Мартинюк Т.Б. Модель порогового нейрона на основе параллельной обработки по разностным срезам // Кибернетика и системный анализ. – 2005. – № 4. – С. 78–79.
  22. Parallel transformation / L.I. Timchenko, M.A. Grudin, T.B. Martynuk et al. // УСиМ. – 1998. – № 5. – С. 93–95.
  23. Пат. 46877 України, G06G7/14, G06F7/50. Конвейерний підсумовуючий пристрій / Т.Б. Мартинюк, В.П. Кожем'яко, А.В. Кожем'яко та ін. – №99063405; Заявл. 18.06.1999; Опубл. 17.06.2002. – Бюл. № 6. – 4 с.
  24. Багатофункціональна регулярна структура для обробки векторних даних / Т.Б. Мартинюк, А. Буда, В.В. Хом'юк та ін. // Праці 7-ої Всеукр. міжнар. конф. УкрОБРАЗ'2004. – Київ. – 2004. – С. 33–36.
  25. Васюра А.С., Мартинюк Т.Б., Кожем'яко А.В. Исследование процесса конвейерной обработки массива чисел // Оптико-електронні інформаційно-енергетичні технології. – 2002. – № 3. – С. 85–94.
  26. Кожем'яко А.В. Систолическі структури для багатооперандної обробки векторних даних: Автореф. дис... канд. техн. наук: 05.13.13 / Вінниц. нац. техн. ун-т. – Вінниця, 2004. – 20 с.
  27. Очин В.Ф. Вычислительные системы обработки изображений. – Л.: Энергоатомиздат, 1989. – 136 с.
  28. Буков А.А. Технические нервные системы. Обучаемые системы управления со зрением для промышленных роботов. – Липецк: Изд-во Липец. техн. ун-та, 2001. – 223 с.
  29. Бернюков А.К., Сушкова Л.Т. Распознавание биоэлектрических сигналов // Зарубежная радиоэлектроника. – 1996. – № 12. – С. 47–51.
  30. Мартинюк Т.Б., Хом'юк В.В. Методи та засоби паралельних перетворень векторних масивів даних. – Вінниця: «УНІВЕРСУМ-Вінниця», 2005. – 203 с.

Поступила 19.09.2008

Тел. для справок: (0432) 262-316, 655-862, 276-422 (Вінниця)

E-mail: [oeipt@vstu.vinnica.ua](mailto:oeipt@vstu.vinnica.ua), [vikira\\_v@mail.ru](mailto:vikira_v@mail.ru)

[kvantron@svitonline.com](mailto:kvantron@svitonline.com)

© Т.Б. Мартинюк, А.В. Кожем'яко, В.В. Хом'юк, 2009