

УДК 681.32.001.

РЕАЛИЗАЦИЯ ГРУППОВЫХ ОПЕРАЦИЙ
НА РОТОРНОМ КОНВЕЙЕРНОМ ПРОЦЕССОРЕ

Я.И. Фет

Эффективное решение проблемы существенного увеличения производительности вычислительных систем требует комплексного подхода к разработке методов решения задач, языков программирования и оборудования машин. При использовании современных языков, особенно крупноблочных языков программирования [1], большую роль играет аппаратная интерпретация [2]. В настоящей работе исследуются возможности реализации некоторых групповых операций на конвейерном вычислительном устройстве специального вида - роторном процессоре [3].

Будем рассматривать следующие групповые операции:

1. Редуктивные операции над численными векторами.

В соответствии с [4] редукцией вектора \underline{x} по любой бинарной функции f называется результат последовательного применения этой функции ко всем компонентам вектора (начиная, например, с правой компоненты). Если $\underline{x} = (x_1, x_2, \dots, x_n)$, то операция редукции записывается следующим образом:

$$f/\underline{x} = x_1 f(x_2 f(\dots x_{n-2} f(x_{n-1} f x_n))).$$

2. Покомпонентные операции над парами векторов.

3. Скалярное произведение векторов.

Предполагается, что компоненты векторов представлены в обычной форме - двоичных чисел с плавающей запятой.

Что касается набора функций, то для редуктивных операций практический интерес представляют функции $+$, $-$, \times , Γ , L ,

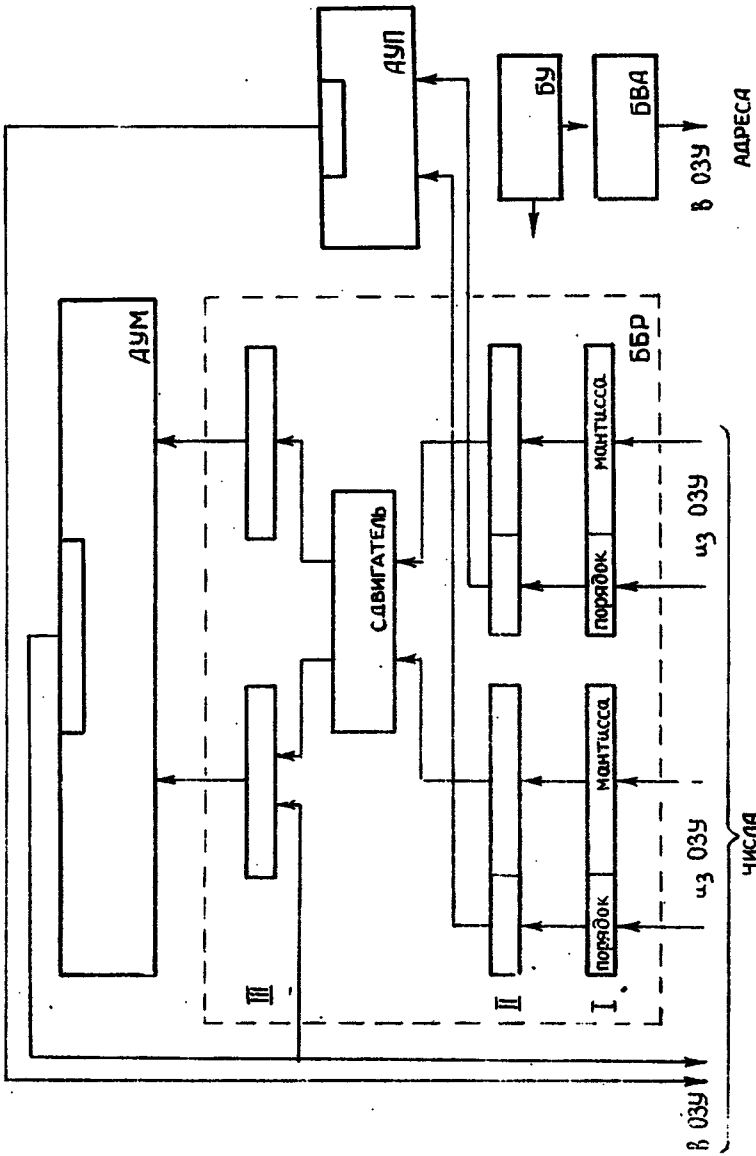


Рис. I

а для покомпонентных, кроме того, операция $x_i + ky_i$ и отношения $<, \leq, =, \neq, \geq, >$.

Теперь перейдем к краткому описанию блок-схемы и принципа работы роторного процессора, после чего покажем возможность эффективной реализации на таком процессоре выделенных выше классов групповых операций.

§ 1. Роторный процессор

Роторный конвейерный процессор (рис. 1) состоит из блока буферных регистров ББР, арифметического устройства порядков АУП, арифметического устройства мантисс АУМ, блока выработки адресов БВА и блока управления БУ.

ББР содержит три пары буферных регистров - регистры I-го, II-го и III-го уровней.

Работа всех блоков процессора совмещена во времени. БВА формирует адреса компонент векторов в соответствии с их размещением в оперативном запоминающем устройстве (ОЗУ). В некотором i -м цикле на буферные регистры I уровня поступают из ОЗУ i -е компоненты обрабатываемых векторов. В это время принятые в $(i - 1)$ -м цикле предыдущие, $(i - 1)$ -е компоненты находятся на буферных регистрах II уровня, и в АУП производится предварительный анализ их порядков. В соответствии с результатами этого анализа АУП в следующем цикле выдаст сигнал управления сдвигателем ББР, обеспечивая необходимый относительный сдвиг мантисс обрабатываемых чисел. Принятые в $(i - 2)$ -м цикле компоненты к этому времени уже прошли предварительную подготовку, и над их мантиссами, находящимися на буферных регистрах III уровня, в АУМ выполняется заданная операция.

Арифметические операции в АУМ выполняются на многоходовом сумматоре роторного типа [5], который имеет два режима работы:

- Режим накапливающего сложения с хранением переносов всех двоичных чисел, поступающих на входы сумматора в последовательных циклах работы. Этот режим используется в процессе выполнения операций с накапливающимся результатом (скалярное произведение, редуктивное сложение и вычитание).
- Режим истинного сложения всех чисел, поступающих на входы

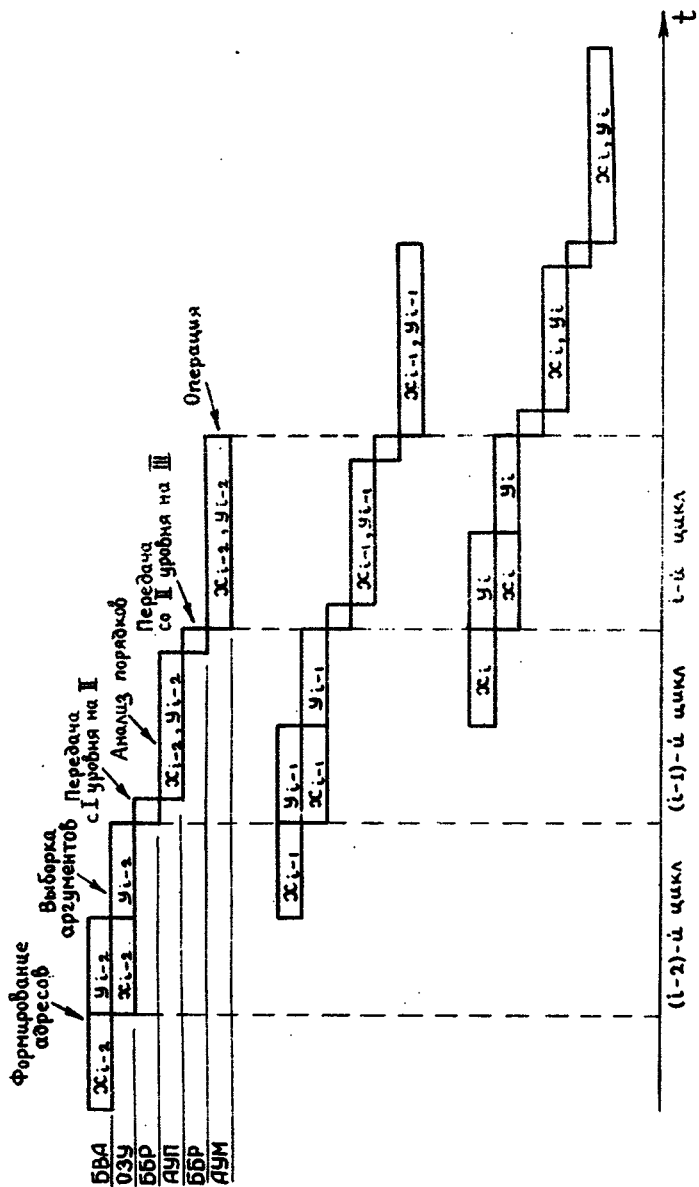


Рис. 2

сумматора в данном цикле. Используется при выполнении покомпонентных операций, а также в заключительной стадии всех других операций.

Результаты покомпонентных операций выдаются в ОЗУ с выходных регистров АУМ и АУП в каждом цикле, а операций с накапливающимся результатом - по окончании группы.

Для того, чтобы обеспечить возможность работы без переполнения в накапливающем режиме при значительных размерностях векторов, в роторном сумматоре предусмотрены дополнительные разряды для целой части результата. В заключительной стадии, после ассимиляции переносов, мантисса результата (если это необходимо) нормализуется вправо, а порядок соответственно подправляется.

§ 2. Особенности выполнения групповых операций

1°. **ВЫЧИСЛЕНИЕ СКАЛЯРНЫХ ПРОИЗВЕДЕНИЙ.** На рис. 2 приведена временная диаграмма работы роторного процессора при вычислении скалярных произведений. Заштрихованными прямоугольниками отмечены периоды работы отдельных блоков. Рядом приведены некоторые словесные пояснения.

Выполняемая в АУМ операция состоит в данном случае в следующем. Мантисса числа, находящегося в левом регистре Π уровня, используется как множимое, а в правом - как множитель. Частичные произведения поступают с левого регистра через соответствующие цепи АУМ на входы роторного сумматора, который работает в накапливающем режиме. При переходе к следующему циклу накопление продолжается, но теперь уже суммируются частичные произведения следующей пары компонент. Таким образом, накапливающийся в регистрах сумматора результат соответствует сумме произведений поступающих на Π уровень чисел.

Из рисунка видно, что практически все блоки процессора загружены полностью. Скорости работы блоков согласованы с временем выборки аргументов из ОЗУ, благодаря чему достигается максимальная эффективность.

2°. **РЕДУКТИВНЫЕ ОПЕРАЦИИ.** При выполнении любой редуктивной операции в каждом цикле необходимо принять из ОЗУ только одно число - очередную компоненту вектора. Вторым аргументом

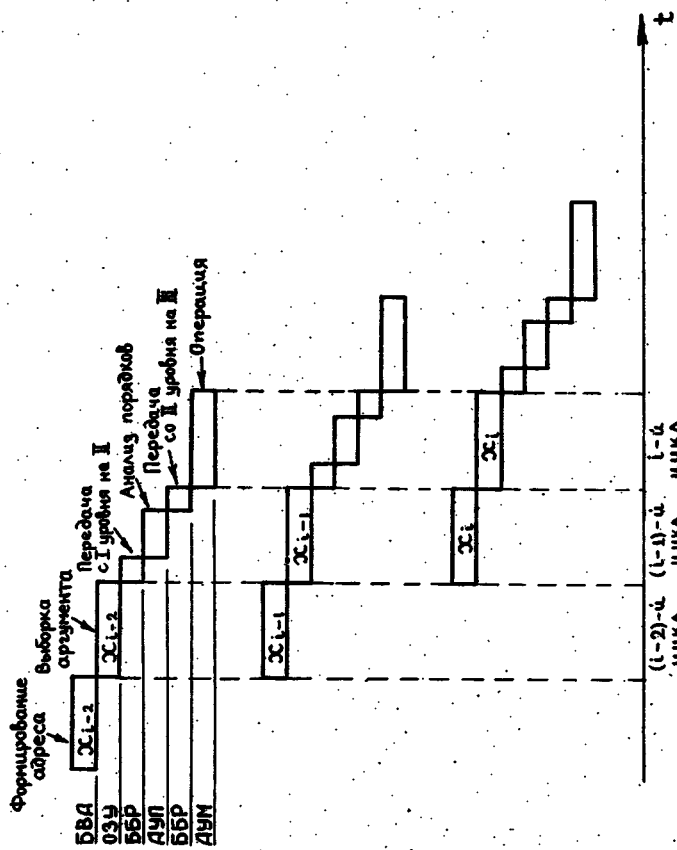


Рис. 3

бинарной операции служит результат предыдущего цикла, хранимый в АУМ. Это позволяет удвоить скорость конвейера по сравнению со скоростью работы при вычислении скалярных произведений, если предъявить повышенные требования к быстродействию его блоков.

Временная диаграмма для этого случая приведена на рис. 3.

Рассмотрим некоторые особенности выполнения различных редуktивных операций.

Редуktивное сложение $+/\mathcal{X}$. Очередные компоненты вектора \mathcal{X} принимаются на левый буферный регистр I уровня. В АУП производится сравнение порядка накопленной суммы с порядком очередной компоненты и вырабатывается соответствующий сигнал управления сдвигом. АУМ в каждом цикле суммирует свое прежнее содержимое с одним новым числом.

Редуktивное вычитание $-/\mathcal{X}$. Аналогично сложению, но перед началом каждой операции в АУМ принудительно меняется знак накопленной суммы. Если обозначить накопленную в i -м цикле сумму через S_i , то при этом:

$$S_3 = -0 + \mathcal{X}_n,$$

$$S_4 = -S_3 + \mathcal{X}_{n-1} = \mathcal{X}_{n-1} - \mathcal{X}_n,$$

$$S_5 = -S_4 + \mathcal{X}_{n-2} = \mathcal{X}_{n-2} - (\mathcal{X}_{n-1} - \mathcal{X}_n)$$

и т.д., что соответствует определению редуktивного вычитания.

Редуktивное умножение \times/\mathcal{X} . Перед началом этой групповой операции на выходной регистр АУМ заносится число $+1$. Очередные компоненты вектора принимаются на правый буферный регистр I уровня. АУП в каждом цикле производит сложение порядка накопленного результата с порядком очередной компоненты. В АУМ в каждом цикле выполняется следующая операция: вычисление истинного результата предыдущего цикла (ассимиляция переносов), пересылка этого результата на левый регистр III уровня и умножение его на содержимое правого регистра III уровня (т.е. на очередную компоненту).

Редуktивное вычисление максимальной компоненты Γ/\mathcal{X} . Мантисса и порядок результата предыдущего цикла находятся соответственно на выходных регистрах АУМ и АУП. Очередная компонента принимается на левый регистр I уровня. В АУП произво-

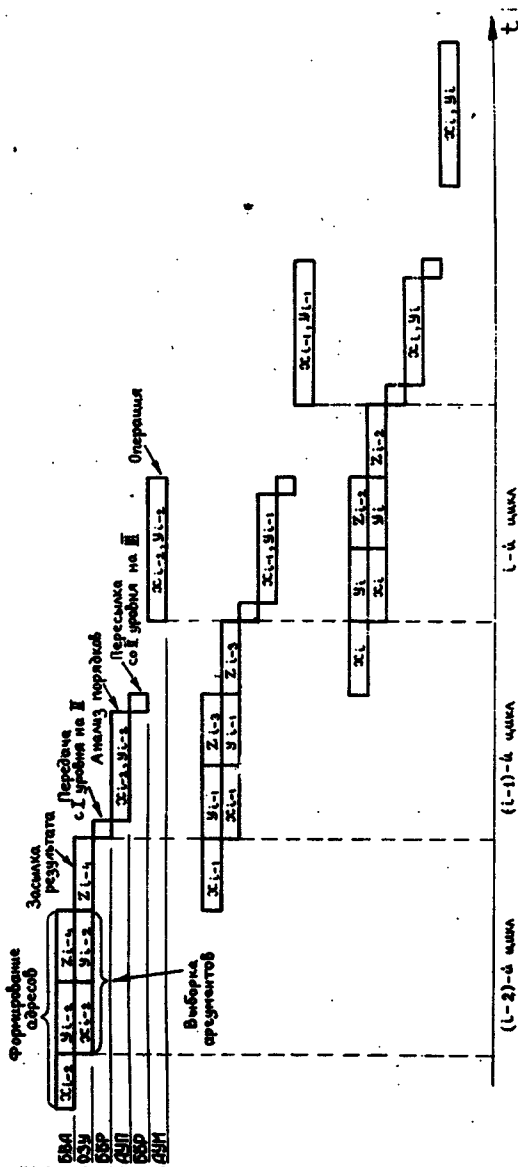


Рис. 4

дятся сравнение порядка результата (P_p) с порядком очередной компоненты (P_k). Если $P_p > P_k$ (предыдущий результат больше нового числа), то все дальнейшие операции данного цикла блокируются за ненадобностью. Если $P_p < P_k$ (новое число больше), то мантисса и порядок нового числа замещают предыдущий результат на выходных регистрах АУМ и АУП. Если $P_p = P_k$, то в АУМ производится сравнение мантиссы предыдущего результата (M_p) с мантиссой новой компоненты (M_k). Здесь остаются две возможности: $M_p \geq M_k$ (предыдущий результат не меньше нового числа) - дальнейшие операции блокируются, либо $M_p < M_k$ (новое число больше) - мантисса нового числа замещает мантиссу предыдущего результата.

Редуктивное вычисление минимальной компоненты L/x . Эта операция аналогична предыдущей и пояснений не требует.

3°. ПОКОМПОНЕНТНЫЕ ОПЕРАЦИИ. Особенностью операций этого класса является необходимость трех обращений к ОЗУ в каждом цикле (два для выборки аргументов и третье - для записи результата). При обычной организации памяти это приводит к некоторому замедлению работы конвейера и частичному простоям оборудования (см. временную диаграмму на рис. 4). Более высокая эффективность может быть достигнута, если ОЗУ позволяет совмещать запись результата с чтением нового аргумента.

Покомпонентное сложение $Z_i = x_i + y_i$. Очередные компоненты вектора x принимаются на левый регистр I уровня, а вектора y - на правый. В АУП производится сравнение порядков x_i и y_i и при передаче со II уровня на III соответствующим образом сдвигается одна из мантисс. Больший из порядков заносится на выходной регистр АУП. В АУМ выполняется истинное сложение чисел, находящихся в левом и правом регистрах III уровня. Результат поступает на выходной регистр АУМ. Выдача очередной компоненты вектора - результата Z с выходных регистров в ОЗУ производится с задержкой на 2 цикла (см. рис. 4).

Покомпонентное вычитание $Z_i = x_i - y_i$ аналогично сложению.

Покомпонентное умножение $Z_i = x_i \times y_i$. Очередные компоненты векторов x и y принимаются соответственно на левый и правый регистры I уровня. В АУП производится сложение порядков, а в АУМ - умножение мантисс (с ассимиляцией переносов).

Операция $Z_i = x_i + k y_i$. При выполнении этой операции

предварительно в арифметическое устройство записывается число k таким образом, что его мантисса поступает в специальный регистр АУМ, а порядок - в специальный регистр АУП.

Очередные компоненты вектора x поступают на левый регистр I уровня, а вектора y - на правый. В АУП производится сложение порядка y_i с порядком k и сравнение полученной суммы (то есть порядка ky_i) с порядком x_i . Большой из этих двух порядков заносится на выходной регистр АУП.

При передаче со II уровня на III соответствующим образом сдвигается мантисса одного из чисел (x_i или y_i).

В АУМ производится умножение мантиссы числа k на мантиссу числа y_i (то есть на содержимое правого регистра III уровня), сложение результата с мантиссой числа x_i (то есть с содержимым левого регистра III уровня) и ассимиляция.

Результаты выдаются с выходных регистров АУМ и АУП так же, как при покомпонентном сложении.

Покомпонентное вычисление максимума $z_i = x_i \vee y_i$. Компоненты принимаются на регистры I уровня. В АУП сравниваются порядки чисел x_i и y_i (Πx_i и Πy_i). Если $\Pi x_i > \Pi y_i$, то в качестве результата на выходные регистры передается число x_i , а если $\Pi x_i < \Pi y_i$, то y_i . В случае $\Pi x_i = \Pi y_i$ в АУМ дополнительно производится сравнение мантисс и на выход АУМ передается большая из них.

Покомпонентное вычисление минимума $z_i = x_i \wedge y_i$. Операция аналогична предыдущей.

Покомпонентное вычисление отношений $u_i = x_i R y_i$. Здесь R - любое из отношений $<, \leq, =, \neq, \geq, >$; u - логический вектор, i -я компонента которого равна единице, если отношение $x_i R y_i$ истинно, и нулю - в противном случае.

Очередные пары компонент поступают, как обычно, на регистры I уровня. В АУП сравниваются их порядки, а в АУМ (в случае равенства порядков) - мантиссы. Специальная логическая схема в соответствии с результатами этих сравнений и с заданным отношением R вырабатывает логическую переменную, которая заносится в очередной разряд выходного регистра АУМ.

§ 3. Выводы

I. Роторный конвейерный процессор реализует групповые операции при сравнительно небольшом расходе оборудования.

Вопрос аппаратной интерпретации групповых операций слабо отражен в литературе.

Из известных устройств для выполнения групповых операций над векторами может быть использован линейный конвейерный процессор, описанный в работе [6]. Если исчислять расход оборудования в "элементарных машинах" (под элементарной машиной в работе [6] понимается устройство для обработки операндов разрядностью в 1 бит), то для линейного конвейера количество элементарных машин имеет порядок n^2 , где n - разрядность обрабатываемых чисел.

В проекте APL - машины [7] предлагается так называемый векторный накопитель, который представляет собой цепочку последовательно включенных скалярных арифметических устройств (то есть обычных параллельных устройств для обработки одной пары чисел). Длина цепочки равна размерности векторов ρV . Количество необходимых элементарных машин имеет порядок $\rho V n$.

В роторном конвейерном процессоре расход оборудования не зависит от размерности векторов. Он определяется в основном сложностью роторного арифметического устройства и наличием трех уровней памяти в ББР. Исходя из опыта проектирования процессора [5], можно считать, что расход элементарных машин имеет порядок $6n$.

2. Конвейерный принцип обработки в сочетании с роторным арифметическим устройством обеспечивает весьма высокое быстродействие, особенно при вычислении скалярных произведений и при выполнении редуцированных операций. Практически это быстродействие ограничено только скоростью чтения аргументов из ОЗУ.

3. В работах [6, 7] предполагается представление чисел с фиксированной запятой. В настоящей статье подробно рассмотрена обработка чисел с плавающей запятой.

4. При программировании задач большого объема возникают существенные трудности с размещением информации в памяти. Использование автономного блока выработки адресов уменьшает жесткость требований к расположению векторов в ОЗУ, облегчая тем самым программирование.

Л и т е р а т у р а

1. Канторович Л.В. Перспективы работ в области автоматизации программирования на базе крупноблочной системы. - "Труды Матем. ин-та АН СССР им. В.А.Стеклова", 1968, 96, с.5 - 15.

2. Глушков В.М. (ред.). Вычислительные машины с развитыми системами интерпретации. Киев, "Наукова думка", 1970.
3. Канторович Л.В., Фет Я.И., Иловайский И.В. Арифметическое устройство цифровой вычислительной машины. — "Бюлл. изобретений", 1968, № 4, авт.свид. 209032.
4. Iverson K.E. A programming language. New York, Wiley, 1962.
5. Иловайский И.В., Фет Я.И. Об одном способе организации арифметического устройства. — "Изв. СО АН СССР", 1966, № 6, вып.2, сер. техн. наук., с.60 - 68.
6. Кухарчук А.Г., Реутов Г.В., Луцкий Г.М. Конвейерный принцип обработки информации. — "Кибернетика", 1968, № 6, с.43-49.
7. Thurber K.J., Muzna J.W. System design of a cellular APL computer. — IEEE Trans. on Comp., 1970, C-19, №4, p.291-301.

Поступила в ред.-изд.отдел
17 февраля 1972 г.