Автоматика и телемеханика. 1999. № 12. С. 155-165.

© 1999 г. В.С. ВЫХОВАНЕЦ, канд. техн. наук
(Приднестровский государственный университет, г. Тирасполь)

ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ ВО ВРЕМЕНИ

На основе анализа комплексной эффективности кратных вычислений обоснована классификация методов их структурной организации. Показано, что параллельные вычисления могут быть разделены на параллельные вычисления в пространстве и параллельные вычисления во времени. Рассмотрены примеры реализации параллельных вычислений во времени на различных уровнях вычислительных средств: программном, командном, операционном и физическом. На операционном уровне распараллеливание во времени основано на параллельной декомпозиции логических функции, представленной в обобщенной полиномиальной форме.

1. Введение

Возможности дальнейшего наращивания производительности вычислительных средств в рамках последовательного принципа обработки данных считаются практически исчерпанными, что обусловлено в основном конечной скоростью распространения сигналов. Поиск решений проблемы повышения производительности идет в направлении развития как нетрадиционных принципов обработки данных [1], так и методов структурной организации вычислений: распараллеливания [2] и конвейеризации [3]. Последние предназначены для выполнения одних и тех же вычислений одновременно на нескольких наборах исходных данных.

Под кратными вычислениями понимается процесс обработки не на одном, а сразу на множестве наборов [4, 5]. Кратные вычисления возникают в виде повторного вызова подпрограммы и распараллеливания вычислительного процесса – на программном уровне, в виде конвейерного и параллельного выполнения команд – на командном уровне, в виде распараллеливания и конвейеризации операций – на операционном уровне.

В прикладных задачах кратные вычисления возникают при параллельном управлении несколькими одинаковыми объектами, находящимися в разных физических состояниях; при моделировании (воспроизведении) системы на нескольких входных воздействиях, например при тестировании и технической диагностике; в системах управления, где требуется быстрая реакция на изменение внешних условий. В последнем случае осуществляется упреждающее вычисление реакции системы на возможные (вероятные) входные наборы, например, находящиеся на единичном расстоянии по Хеммингу, а при поступлении очередного набора уже имеется требуемый результат.

Настоящая статья посвящена исследованию кратных вычислений на различных уровнях вычислительных средств. Под кратными понимаются вычисления, осуществляемые сразу (одновременно) на нескольких наборах исходных данных. Суть подхода заключается в том, что кратные вычисления представляются в виде композиции известных форм однократных вычислений. Для исследования последних вводятся критерии эффективности, учитывающие как временные затраты, так и сложность используемых аппаратных средств. На основе анализа относительной эффективности обосновывается классификация методов структурной организации кратных вычислений, а использование абсолютных критериев позволяет учесть и эффективность форм описания вычисления в однократном случае.

2. Критерии эффективности

Для оценки эффективности кратных вычислений введем следующие критерии: эффективность по времени вычисления ξ, эффективность по сложности вычислений η и комплексную эффективность ϑ, равную произведению эффективностей по времени ξ и сложности η.

Эффективность по времени ξ определяет во сколько раз время q-кратных вычислений τ в пересчете на один набор данных меньше, чем время вычислений в однократном случае, которое принято за единицу измерения времени:

$$\xi = \frac{1}{\tau/q} = \frac{q}{\tau}.$$

Эффективность по сложности вычислений η определяет во сколько раз аппаратурные (пространственные) затраты ν при кратных вычислениях меньше аналогичных затрат в однократном случае:

$$\eta = \frac{1}{\nu}.$$

Здесь, как и ранее, пространственные затраты в однократном случае приняты за единицу измерения сложности вычислений.

Комплексная эффективность ω определяет какой ценой достигается повышение производительности и равна времени вычисления на одном наборе данных в пересчете на единицу оборудования:

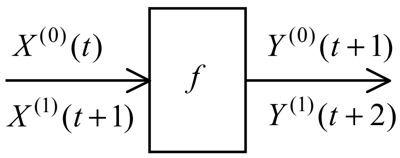
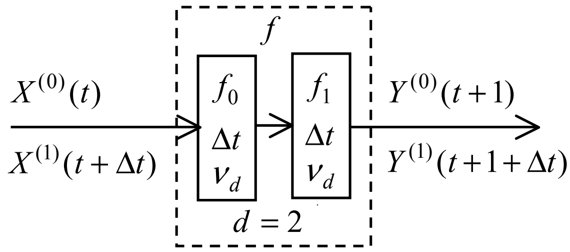
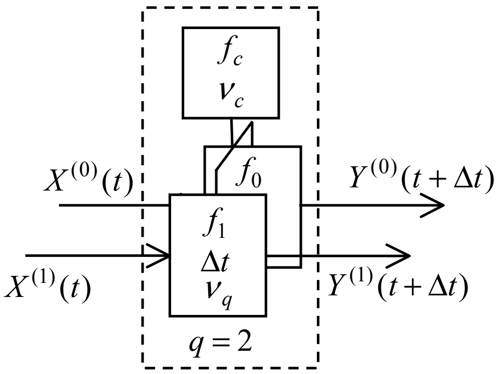
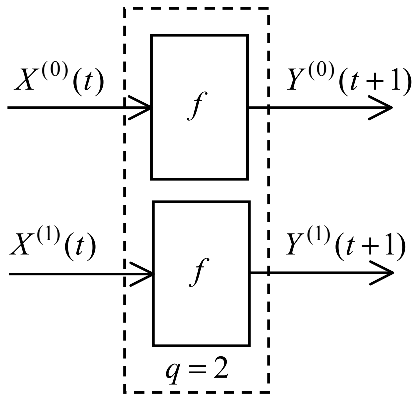
$$\omega = \xi\eta = \frac{q}{\tau\nu}. \qquad (1)$$

Единичная комплексная эффективность характеризует экстенсивный путь повышения производительности вычислительного средства, а больше единицы - интенсивный.

3. Структурная организация кратных вычислений

В основу классификации методов структурной организации кратных вычислений положим принципы повторности: параллельная или последовательная повторность, повторность в пространстве или во времени. На основе введенных классификационных признаков можно выделить параллельные и последовательные вычисления, осуществляемые как в пространстве, так и во времени (Табл. 1).

Таблица 1. Структурная организация кратных вычислений

	Во времени	В пространстве
Последовательно	 $\omega_1 = 1$	 $\omega = \dfrac{q}{d+q-1}\dfrac{1}{\nu\,\Delta t} > 1$
Параллельно	 $\omega_4 = \dfrac{q}{\nu_c + q\nu_q}\dfrac{1}{\Delta t} > 1$	 $\omega_2 = 1$

При последовательных вычислениях во времени (повторная обработка) различные наборы данных $X^{(t)}$ ($t=\overline{0,q-1}$), подаются на вход вычислительного средства в разные моменты времени t, отстоящие друг от друга на время однократного вычисления, т.е. последовательность входных наборов

$$X^{(0)}(t), X^{(1)}(t+1), \ldots, X^{(q-1)}(t+q-1)$$

порождает соответствующую последовательность результатов $Y^{(t)}$ ($t=\overline{0,q-1}$),

$$Y^{(0)}(t+1), Y^{(1)}(t+2), \ldots, Y^{(q-1)}(t+q),$$

а комплексная эффективность может быть определена так:

$$\xi_1 = \frac{q}{q} = 1, \quad \eta_1 = \frac{1}{1} = 1, \quad \omega_1 = \xi_1 \eta_1 = 1$$

При параллельных вычислениях в пространстве (распараллеливание или многоэлементная обработка) используется несколько вычислительных средств, каждое из которых выполняет вычисление на своем наборе. В этом случае результаты появляются на выходе через время однократного вычисления, но при этом требуется дублирование оборудования, равное кратности вычислений m. Входные наборы данных

$$X^{(0)}(t), X^{(1)}(t), \ldots, X^{(q-1)}(t)$$

подаются одновременно и одновременно появляются на выходе соответствующие результаты

$$Y^{(0)}(t+1), Y^{(1)}(t+1), \ldots, Y^{(q-1)}(t+1),$$

Комплексная эффективность многоэлементной обработки, как и при последовательных вычислениях во времени, будет равна единице:

$$\xi_2 = \frac{q}{1} = q, \quad \eta_2 = \frac{1}{q}, \quad \omega_2 = \xi_2 \eta_2 = 1.$$

При последовательных вычислениях в пространстве (конвейеризация или многостадийная обработка) различные наборы данных подаются на вход также в разные моменты времени, но отстоящие друг от друга на время Δt, которого достаточно для вычислений, осуществляемых самой медленной из d ступеней конвейера. В этом случае последовательность входных наборов

$$X^{(0)}(t), X^{(1)}(t+\Delta t), \ldots, X^{(q-1)}(t+(q-1)\Delta t)$$

порождает соответствующую последовательность результатов

$$Y^{(0)}(t+d\Delta t), Y^{(1)}(t+(d+1)\Delta t), \ldots, Y^{(q-1)}(t+(d+q-1)\Delta t),$$

откуда получаем выражение для комплексной эффективности:

$$\xi_3 = \frac{q}{(d+q-1)\Delta t}, \quad \eta_3 = \frac{1}{\nu}, \quad \omega_3 = \xi_3 \eta_3 = \frac{q}{d+q-1} \frac{1}{\nu \, \Delta t},$$

где ν – пространственные затраты на конвейерных вычислений, равная сумме затрат на каждой ступени, $\nu = \nu_0 + \nu_1 + \ldots + \nu_{d-1}$. На практике стремятся к такой реализации многостадийной обработки, при которой $\nu \approx 1$, а время вычисления на каждой из стадий $\Delta t \approx 1/d$. Тогда при достаточно большом q имеем

$$\omega_3 \approx \frac{dq}{d+q-1} \sim d,$$

где знаком ~ обозначено асимптотическое равенство, получаемое при q стремящемся к бесконечности.

Параллельные вычисления во времени соответствуют такой организации вычислений, когда на вход вычислительного средства подаются сразу все m наборов

$$X^{(0)}(t), X^{(1)}(t), \ldots, X^{(q-1)}(t)$$

и через некоторое время Δt получаются все q результатов

$$Y^{(0)}(t+\Delta t), Y^{(1)}(t+\Delta t), \ldots, Y^{(q-1)}(t+\Delta t),$$

Очевидно, что для отличия параллельных вычислений во времени от параллельных вычислений в пространстве (по аналогии с последовательными вычислениями во времени и в пространстве) комплексная эффективность

$$\omega_4 = q\frac{1}{\nu\,\Delta t} \tag{2}$$

должна быть больше единицы, где ν – пространственные затраты на параллельные вычисления во времени, а исходные данные (результаты) подаваться (сниматься) одновременно.

Если предположить, что при организации параллельных вычислений во времени $\Delta t = 1$, то получение комплексной эффективности $\omega_4 > 1$ возможно только за счет снижения пространственных затрат. Как видно из (2) это имеет место, когда $\nu < q$, т.е. когда пространственные затраты параллельных вычислений во времени будут меньше чем аналогичные затраты при параллельных вычислениях в пространстве:

$$v = \nu_c + q\nu_q < q,$$

где ν_c – пространственные затраты на реализацию некоторой общей часть оборудования, используемой для нескольких параллельных каналов обработки данных, каждый из которых реализуется с пространственными затратами ν_q.

4. Параллельная декомпозиция

В настоящее время наибольшее распространение получили вычислительные устройства дискретного действия. Математическим аппаратом, используемом при проектировании таких устройств, является алгебра логики и теория автоматов [6].

Дискретная обработка данных представляется в виде дискретной функции $f(X)$, существенно зависящей от n переменных $X = \{x_0, x_1, \ldots, x_{n-1}\}$, где значности переменных x_i, в общем случае, равны k_i ($i = \overline{0, n-1}$), а значность функции f равна некоторому числу k_f, т.е. $x_i \in N_{k_i}$ и $f \in N_{k_f}$, где $N_k = \{0, 1, \ldots, k-1\}$.

Известные методы вычислений своей структурой повторяют формульное описание функции, а их эффективность зависит от компактности такого описания [7]. При последовательных вычислений во времени дискретная функция представляется формулой в некотором функционально полном базисе операций. Количество операций, необходимых для вычисления значения функции определяет эффективность найденного формульного представления.

Для конвейерных вычислений функция $f(X)$ декомпозируется на d подсистем f_j ($j=\overline{0,d-1}$), таким образом, чтобы результат вычисления предыдущей подсистемы f_j являлся аргументами для вычисления следующей подсистемы f_{j+1}:

$$f(X)=f_{d-1}(f_{d-2}(\ldots f_1(f_0(X))\ldots)). \quad (3)$$

При параллельных вычислениях в пространстве формульное описание функции повторяется m раз, что равно кратности вычислений. При параллельных вычислениях во времени комплексная эффективность должна быть больше единицы, что требует более компактного представления функции, заданной сразу на m наборах переменных.

Последнее получим путем объединения формульных описаний функции на каждом наборе в одно выражение с целью его упрощения путем выделения общих подвыражений, участвующих при вычислениях на каждом из наборов. Такое представление назовем ***параллельной декомпозицией*** функции, в отличие от последовательной (3), необходимой для конвейерной обработки.

Так как параллельная декомпозиция основана на форме функции для однократных вычислений, предварительно рассмотрим методику синтеза таких форм.

5. Синтез полиномиальных форм

Полиномиальные формы получили широкое распространение для представления логических функций. В Табл. 2 приведены примеры полиномиальных форм булевых функций [8], где знаки $.\&$, $\vee$, и $\oplus$ обозначают соответственно булевы операции конъюнкции, дизъюнкции и неэквиваленции, а $i=(i_0\, i_1 \ldots i_{n-1})_2$ – представление натурального числа i в двоичной позиционной системе счисления, i_j – двоичные цифры. Известна также арифметическая форма в k-значной логике [9]:

$$f(X)=\sum_{i=0}^{k^n-1} a_i x_0^{i_0} x_1^{i_1} \ldots x_{n-1}^{i_{n-1}},$$

в которой i представляется в k-ичной системе счисления, $i=(i_0\, i_1 \ldots i_{n-1})_k$, а x_i^j обозначает переменную x_i, возведенную в арифметическую степень j.

Автоматика и телемеханика. 1999. № 12. С. 155-165.

Таблица 2. Полиномиальные формы булевых функций

Форма	Аналитическая конструкция	Степенная операция
Дизъюнктивная	$f(X)=\bigvee_{i=0}^{2^n-1} a_i \,\&\, x_0^{i_0} \,\&\, x_1^{i_1} \,\&\ldots\&\, x_{n-1}^{i_{n-1}}$	$x_i^j = \begin{cases} \bar{x}_i, & j=0; \\ x_i, & j=1 \end{cases}$
Конъюнктивная	$f(X)=\mathop{\&}_{i=0}^{2^n-1} a_i \vee x_0^{i_0} \vee x_1^{i_1} \vee\ldots\vee x_{n-1}^{i_{n-1}}$	$x_i^j = \begin{cases} x_i, & j=0; \\ \bar{x}_i, & j=1 \end{cases}$
Жегалкина	$f(X)=\bigoplus_{i=0}^{2^n-1} a_i \,\&\, x_0^{i_0} \,\&\, x_1^{i_1} \,\&\ldots\&\, x_{n-1}^{i_{n-1}}$	$x_i^j = \begin{cases} 1, & j=0; \\ x_i, & j=1 \end{cases}$
Арифметическая 1	$f(X)=\sum_{i=0}^{2^n-1} a_i (x_0^{i_0} \,\&\, x_1^{i_1} \,\&\ldots\&\, x_{n-1}^{i_{n-1}})$	$x_i^j = \begin{cases} 1, & j=0; \\ x_i, & j=1 \end{cases}$
Арифметическая 2	$f(X)=\sum_{i=0}^{2^n-1} a_i (x_0^{i_0} \vee x_1^{i_1} \vee\ldots\vee x_{n-1}^{i_{n-1}})$	$x_i^j = \begin{cases} 0, & j=0; \\ x_i, & j=1 \end{cases}$
Арифметическая 3	$f(X)=\sum_{i=0}^{2^n-1} a_i (x_0^{i_0} \oplus x_1^{i_1} \oplus\ldots\oplus x_{n-1}^{i_{n-1}})$	$x_i^j = \begin{cases} 0, & j=0; \\ x_i, & j=1 \end{cases}$
Уолша	$f(X)=\sum_{i=0}^{2^n-1} a_i x_0^{i_0} x_1^{i_1} \ldots x_{n-1}^{i_{n-1}}$	$x_i^j = \begin{cases} 1, & j=0; \\ (-1)^{x_i}, & j=1 \end{cases}$

Обобщенную полиномиальную форму, синтезируемую в фундаментальной алгебре (конечном поле или коммутативном кольце без делителей нуля) при различных значностях переменных, представим в виде:

$$f(X)=\sum_{i=0}^{m-1}(x_0^{i_0}\circ_0 x_1^{i_1}\circ_1\ldots\circ_{n-2} x_{n-1}^{i_{n-1}}\circ_{n-1} c_i)\times a_i=\sum_{i=0}^{m-1}\theta_i(X)\times a_i\,, \qquad (4)$$

где $m=k_0k_1\ldots k_{n-1}$; a_i – коэффициенты формы; c_i – произвольные константы; $\circ_j$ – соединительные операции; $x_i^{i_j}$ – переменная x_i в степени i_j; $i=(i_0\,i_1\ldots i_{n-1})_m$ – представление числа i в системе счисления при различных основаниях цифр, равных k_0, k_1, …, k_{n-1}; θ_i – полиномиальные ортогональные функции.

Умножение a_i на θ_i и суммирование полученных значений осуществляем в одной из фундаментальных алгебр, например по модулю $k\ge k_f$, где k_f – значность представляемой функции. Порядок выполнения операций при вычислении θ_i слева направо с приоритетом операции возведения в степень, а сами операции таковы, что область значений предыдущей операции совпадает или включается в область определения следующей.

Методику синтеза формы (4) определим на основе дискретного ортогонального преобразования [10]:

$$\begin{cases} \mathbf{F} = \mathbf{D} \times \mathbf{A}; \\ \mathbf{A} = \mathbf{Q} \times \mathbf{F}, \end{cases}$$

где $\mathbf{F}$ – характеристический вектор функции f, $\mathbf{A}$ – вектор коэффициентов формы, $\mathbf{D}$ и $\mathbf{Q}$ – матрицы прямого и обратного преобразования размерности $m \times m$. Последние получаем следующим образом.

1). Задаем ядро преобразования, которое определяет степенные операции для каждой переменной x_t и соответствующие матрицы $\mathbf{\Gamma}_t$ размерностью $k_t \times k_t$ с элементами:

$$\gamma_t(i, j) = i^j \quad (i, j = \overline{0, k_t - 1},\ t = \overline{0, n-1}),$$

где i (j) – номер строки (столбца). Строки (столбцы) матрицы $\mathbf{\Gamma}_t$ должны быть линейно независимы.

2). Строим матрицу $\mathbf{D}$ по рекуррентному правилу:

$$\mathbf{D}_0 = \mathbf{\Gamma}_0, \quad \mathbf{D}_{t+1} = \mathbf{D}_t \otimes_t \mathbf{\Gamma}_t \ \ (t = \overline{0, n-2}), \quad \mathbf{D} = \mathbf{D}_{n-1} \circ_{n-1} \mathbf{C}, \tag{5}$$

где $\mathbf{C}$ – матрица констант, состоящая из m одинаковых строк m произвольных констант из области определения $\circ_{n-1}$; $\otimes_t$ – обобщенная операция кронекеровского произведения матриц, которую определим следующим образом:

$$\mathbf{D}_t \otimes_t \mathbf{\Gamma}_t = [\mathbf{D}_t \circ_t \gamma_t(i, j) \mid i, j = \overline{0, k_t - 1}]. \tag{6}$$

В результате получаем матрицу, состоящую из k_t^2 подматриц $\mathbf{D}_t$, каждая из которых поэлементно преобразована операцией $\circ_t$ с правым операндом $\gamma_t(i, j)$. Выбор операций и их последовательности не должны вести к линейной зависимости срок (столбцов) промежуточных матриц $\mathbf{D}_t$, а константы c_i задаем так, чтобы определитель $\mathbf{D}$ был отличен от нуля;

3). Матрицу $\mathbf{Q}$ определяем из условия ортогональности: $\mathbf{Q} \times \mathbf{D} = \mathbf{E}$, где $\mathbf{E}$ – единичная матрица размерности $m \times m$.

Вычисление определителей при обращении $\mathbf{D}$, а также проверка линейной зависимости строк (столбцов) выполняем в поле (целостном кольце) используемых в выражении (4) операций сложения и умножения. Определение функциональной полноты базиса, таким образом, сводим к проверке линейной независимости строк (столбцов) при построении матрицы $\mathbf{D}$.

Традиционно вычисления в многозначной логике сводятся к вычислениям в булевой алгебре путем представления многозначных переменных совокупностью булевых значений. Смешанная значность переменных и расширенный базис операций позволяют, при прочих равных условиях, получить эффективные (компактные) представления для большего количества функций. Для каждой функции можно подобрать некоторые степенные и соединительные операции, при использовании которых получаемая форма будет иметь требуемые полезные свойства, например, много нулевых коэффициентов, наименьший объем памяти для их хранения, наименьшее время вычислений, возможность факторизации матрицы дискретного преобразования, и т.д.

6. Абсолютная эффективность

Как было показано выше, однократные вычисления функции можно производить по нескольким ее представлениям (4) и, как следствие этого, с различной эффективностью. Введенные ранее относительные критерии эффективности необходимы для структурной классификации кратных вычислений и не учитывают эффективность представления самой функции. Следовательно необходимо использование абсолютных критериев, учитывающих эффективность однократных вычислений.

Расчет абсолютных эффективностей будем осуществлять относительно табличного способа вычислений, который инвариантен выбору функции. Пространственная эффективность табличного способа η_a определяется объемом памяти ν_t для хранения таблицы истинности функции f, а эффективность по времени вычисления ξ_a – временем извлечения данных τ_t из этой таблицы:

$$\xi_a = \frac{\tau_t}{\tau}, \quad \eta_a = \frac{\nu_t}{\nu}, \quad \omega_a = \xi_a \eta_a = \frac{\tau_t \nu_t}{\tau \nu}, \tag{7}$$

где τ и ν – соответственно временные и пространственные затраты на реализацию однократных вычислений исследуемым способом.

Использование абсолютных критериев позволяет не только сравнивать различные представления одной и той же функции, но и различные представления произвольных функций.

7. Распараллеливание во времени

Для параллельной декомпозиции полиномиальную форму функции (4) записываем для каждого набора переменных $X^{(t)}$ ($t = \overline{0, q-1}$):

$$f(X^{(t)}) = \sum_{i=0}^{m-1} (x_{0t}^{i_0} \circ_0 x_{1t}^{i_1} \circ_1 \ldots \circ_{n-2} x_{n-1\,t}^{i_{n-1}} \circ_{n-1} c_i) \times a_i = \sum_{i=0}^{m-1} \theta_i (X^{(t)}) \times a_i \,, \tag{8}$$

где $x_{i\,t}$ – значение переменной x_i из t-го набора.

Из выражения (8) видно, что для каждого набора переменных общими подвыражениями являются коэффициенты a_i, константы c_i и степени переменных i_j. Следовательно блоки, реализующие эти подвыражения могут быть общими для всех параллельных каналов. Так как аппаратные затраты растут не кратно q, комплексная эффективность вычислений будет больше единицы.

Структура логического процессорного элемента, реализующего распараллеливание во времени приведена на рис. 1, где для двух параллельных каналов выделено общее оборудование: блок степеней переменных **I** и блоки хранения коэффициентов **A** и **C**. Вычисление логической части полинома θ_i выполняется параллельно в пространстве путем использования поразрядных (векторных) операций. Коэффициенты полинома **A** определяют вычисляемую функцию и выступают в роли исполняемой программы (задают алгоритм обработки данных).

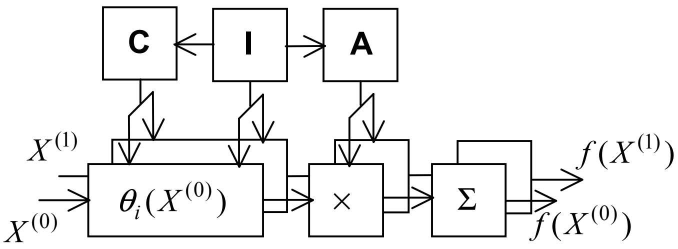


Рис. 1. Параллельные вычисления во времени

На рис. 2 приведена структура параллельно-конвейерного процессорного элемента, отличающаяся от предыдущей тем, что вычисление полиномиальных функций, умножение на коэффициенты и сложение полученных величин осуществляется на конвейере с общей глубиной $d+2$. Тем самым совмещены два метода структурной организации кратных вычислений – параллельные вычисления во времени и последовательные вычисления в пространстве. Это обеспечивает наивысшую комплексную эффективность, равную произведению эффективностей каждого из методов.

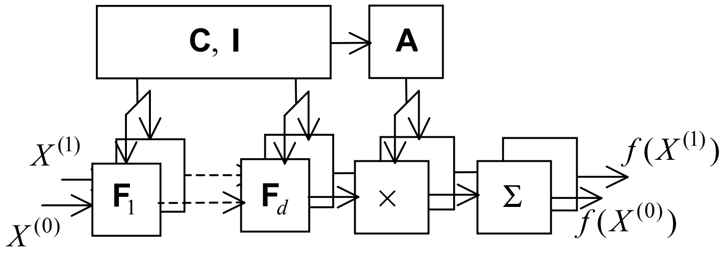


Рис. 2. Параллельно-конвейерный процессорный элемент

Заметим, что для систем логических функций параллельная декомпозиция может быть осуществлена в форме кратных вычислений. Методика синтеза таких форм основана на результатах, полученных В.Д. Малюгиным [4] и рассмотрена в статье [5].

8. Демонстрационный пример

Представим в полиномиальной форме дискретную функцию, заданную таблицей истинности (табл. 3). Из таблицы видно, что переменная x_0 имеет значность 2, x_1 – значность 3, а сама функция – значность 4. В соответствии с (4) искомая полиномиальная форма на кольце вычетов по модулю 4, задаваемом арифметическими операциями сложения и умножения по модулю 4, будет иметь вид:

$$f(X^{(t)}) = \sum_{i=0}^{5} (x_0^{i_0} \circ_0 x_1^{i_1} \circ_1 c_i) \times a_i ,$$

где $\circ_0$ определим как сдвиг битового представления левого операнда вправо на число двоичных разрядов, задаваемое правым операндом, а степенные операции зададим как битовые операции неэквиваленции и дизъюнкции:

$$\mathbf{\Gamma}_0 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{\Gamma}_1 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 1 & 3 \\ 2 & 3 & 2 \end{bmatrix}.$$

Использование в рассматриваемом примере битовых операций определяется их эффективной реализацией на современных вычислительных средствах с двоичным кодированием данных.

Таблица 3. Дискретная функция

x_0	x_1	$f(x_0, x_1)$
0	0	2
1	0	1
0	1	3
1	1	3
0	2	0
1	2	1

Далее, в соответствии с выражением (5) и (6), найдем матрицу $\mathbf{D}_1$,

$$\mathbf{D}_1=\begin{bmatrix}0&1\\1&0\end{bmatrix}\otimes_0\begin{bmatrix}0&1&2\\1&1&3\\2&3&2\end{bmatrix}=\left[\begin{array}{cc:cc:cc}0&0&1&0&2&1\\0&0&0&1&1&2\\\hdashline 1&0&1&0&3&1\\0&1&0&1&1&3\\\hdashline 2&1&3&1&2&1\\1&2&1&3&1&2\end{array}\right],$$

определитель которой по модулю 4 равен нулю. В связи с чем определим операцию $\circ_1$ как операцию максимум и выберем вектор коэффициентов $\mathbf{C}=[3\,0\,0\,0\,0\,0]$. Это обеспечит обращаемость матрицы прямого преобразования $\mathbf{D}$ на кольце вычетов по модулю 4:

$$\mathbf{D}=\begin{bmatrix}3&0&0&0&0&0\\3&0&0&0&0&0\\3&0&0&0&0&0\\3&0&0&0&0&0\\3&0&0&0&0&0\\3&0&0&0&0&0\end{bmatrix}\circ_1\mathbf{D}_1=\left[\begin{array}{cc:cc:cc}3&0&1&0&2&1\\3&0&0&1&1&2\\\hdashline 3&0&1&0&3&1\\3&1&0&1&1&3\\\hdashline 3&1&3&1&2&1\\3&2&1&3&1&2\end{array}\right],$$

$$\mathbf{Q}=\frac{1}{2}\begin{bmatrix}3&3&1&1&1&1\\3&1&3&1&3&1\\0&2&0&0&0&2\\3&3&1&3&3&3\\2&0&2&0&0&0\\1&1&1&1&1&3\end{bmatrix}.$$

Умножая матрицу $\mathbf{Q}$ на характеристический вектор функции $\mathbf{F}=[2\,1\,3\,3\,0\,1]^T$ получаем вектор коэффициентов формы $\mathbf{A}$:

$$\mathbf{A}=\mathbf{Q}\times\mathbf{F}=\frac{1}{2}\begin{bmatrix}3&3&1&1&1&1\\3&1&3&1&3&1\\0&2&0&0&0&2\\3&3&1&3&3&3\\2&0&2&0&0&0\\1&1&1&1&1&3\end{bmatrix}\times\begin{bmatrix}2\\1\\3\\3\\0\\1\end{bmatrix}=\begin{bmatrix}0\\2\\0\\0\\1\\0\end{bmatrix},$$

по которому конструируем искомую полиномиальную форму:

$$f(X)=(x_0^1\circ_0 x_1^0)\times 2+(x_0^0\circ_0 x_1^2)\times 1=(x_0^1\circ_0 x_1)\times 2+(x_0\circ_0 x_1^2),$$

где учтено, что $x_0^0=x_0$ и $x_1^0=x_1$.

Рассчитаем абсолютную эффективность найденной формы. Из последнего выражения следует, что для вычисления функции необходимо выполнить 6 операции: сложение, умножение на 2, два сдвига, дизъюнкцию и неэквиваленцию. На языке программирования C вычисление функции f будет иметь вид:

$$f = ((x_1 >>! x_0) << 1) + ((x_1 \mid 2) >> x_0) .$$

При табличных вычислениях значение функции найдем из выражения, содержащего 3 операции:

$$f = table[(x_1 << 1) + x_0],$$

где $table$ – таблица истинности функции (массив длиной 6 байт). В итоге в соответствии с (7) получаем искомые абсолютные эффективности:

$$\xi_a = \frac{\tau_t}{\tau} = \frac{3}{6} = 0{,}5 , \quad \eta_a = \frac{\nu_t}{\nu} = \frac{6\log_2 k_f}{2\log_2 k_f} = 3 , \quad \omega_a = \xi_a\eta_a = 1{,}5 ,$$

где k_f – значность функции.

В свою очередь комплексную эффективность двукратных вычислений, осуществляемых параллельно во времени, подсчитаем следующим образом. На 6 выполняемых операций необходим блок памяти для хранения 2 коэффициентов. Откуда долю общего оборудования определим как $2/6 \approx 0{,}33$ и в соответствии с (1) имеем:

$$\xi = \frac{2}{1} = 2 , \quad \eta = \frac{1}{2 - 0{,}33} \approx 0{,}6 , \quad \omega = \xi\eta \approx 1{,}2 .$$

9. Заключение

Дадим расширенную интерпретацию полученных результатов. На операционном уровне реализация параллельных вычислений во времени основана на параллельной декомпозиции функции, которая заключается в выделении общих выражений, используемых для вычисления на всех наборах переменных и может быть выполнена на основе представления функции в обобщенной полиномиальной форме. Примером распараллеливания во времени на программном уровне может служить многопроцессорная ЭВМ с общей памятью, на уровне команд – процессор Intel Pentium, у которого для нескольких каналов обработки данных имеется общая кэш-память команд и данных [11].

Распараллеливание во времени возможно также на физическом уровне, например при параллельной оптической обработке, осуществляемой устройством на рис. 3.

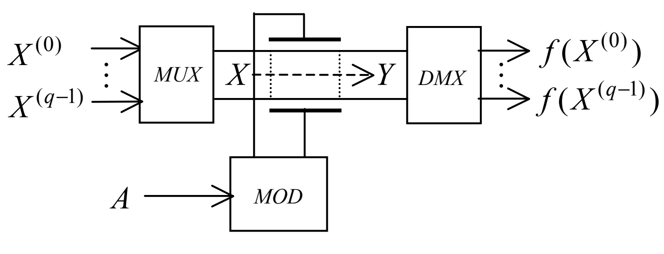


Рис. 3. Параллельная во времени оптическая обработка данных

По некоторому каналу распространяется совокупность сигналов $X^{(t)}$, имеющих разную поляризацию или разные моды распространения, что обеспечивает отсутствие их существенного влияния друг на друга.

В среду обработки сигналы вводятся с помощью пространственного мультиплексора *MUX*, а после обработки разделяются пространственным демультиплексором *DMX*.

В среде обработки все сигналы подвергаются одинаковому моделирующему воздействию со стороны обрабатывающего элемента, являющегося частью среды распространения (показан пунктирной линией). Моделирующий сигнал выдается пространственным модулятором *MOD*, на вход которого подается обрабатывающий сигнал *A*, задающий закон (алгоритм) преобразования.

В рассмотренном примере, помимо наличия общего оборудования для параллельных каналов, сами каналы не дублируются, а используется один общий канал, в котором обрабатывается несколько сигналов одновременно.

Литература

1. *Амамия М., Танака Ю.* Архитектура ЭВМ и искусственный интеллект. М.: Мир, 1993.

2. *Головкин Б. А*. Параллельные вычислительные системы. М.: Наука, 1980.

3. *Кухарчук А. Г., Луцкий Г. М.* Конвейерный принцип обработки информации // Кибернетика. 1968. № 6. С. 43-49.

4. *Малюгин В. Д.* Параллельные логические вычисления посредством арифметических полиномов. М.: Наука, 1997.

5. *Выхованец В. С., Малюгин В. Д.* Кратные логические вычисления // Автоматика и телемеханика. 1998. № 6. С.163-171.

6. *Гаврилов М. А., Девятков В. В., Пупырев Е. И.* Логическое проектирование дискретных автоматов. М.: Наука, 1977.

7. *Глушков В. М*. Синтез цифровых автоматов. М.: Физматгиз, 1962.

8. *Кухарев Г. А., Шмерко В. П., Янушкевич С. Н.* Техника параллельной обработки данных на СБИС. Минск: Вышэйшая школа, 1991.

9. *Антоненко В. М., Иванов А. А., Шмерко В. П.* Линейные арифметические формы k-значных логик и их реализация на систолических массивах // Автоматика и телемеханика. 1995. № 4. С. 139-155.

10. *Кухарев Г. А., Шмерко В. П.* Новые возможности дискретного преобразования Фурье для аналитического описания бинарных и многозначных данных // Распознавание,

Автоматика и телемеханика. 1999. № 12. С. 155-165.

классификация, прогноз. Математические методы и их применение / ВЦ АН СССР. М.: Наука, 1993. Вып. 3. С. 112-147.

11. *Шпаковский Г. И.* Организация параллельных ЭВМ и суперскалярных процессоров. Минск: Белгосуниверситет, 1996.