4.1. КАНОНИЧЕСКАЯ ФУНКЦИОНАЛЬНАЯ СТРУКТУРА КОНВЕЙРНОГО ПРОЦЕССОРА

В конвейерных вычислительных системах основной объем операций по обработке данных выполняется одним или несколькими конвейерными процессорами (или кратко: конвейерами). Конвейеры оперируют с векторами данных. Такой *вектор суть одномерный массив или одномерная упорядоченная совокупность элементов данных одного типа*. Если воспользоваться терминами алгебры матриц, то вектор данных – это или столбец, или строка, или диагональ двумерной матрицы, либо матрица-столбец или матрица-строка вида:

$$\boldsymbol{A} = \|A_1, A_2, ..., A_i, ..., A_n\| = (A_1, A_2, ..., A_i, ..., A_n),$$

где A_i есть i-ый компонент (или элемент, или элемент-операнд, или скалярная величина, или просто "скаляр" или число), $i = \overline{1,n}$.

В конвейере векторные операции реализуются аппаратурно, поэтому его называют также *векторным процессором*. При этом всегда предусматриваются операции покомпонентного сложения и покомпонентного умножения двух векторов, а также либо покомпонентное деление векторов, либо формирование вектора из чисел, обратных компонентам данного вектора. Могут иметься также векторные команды для более сложных операций (например, покомпонентного извлечения квадратного корня). В конвейеры может быть заложена возможность реализации "триад" (linked triad – "сцепленных триад"), т.е. операций вида

$$\boldsymbol{A} + \alpha\boldsymbol{B},$$

где $\boldsymbol{A}$ и $\boldsymbol{B}$ – векторы данных, α – скаляр, $\alpha\boldsymbol{B}$ – вектор, компоненты которого равны соответствующим компонентам $\boldsymbol{B}$, умноженным на α. Возможны и другие разновидности триады, например:

$$(\boldsymbol{A} + \alpha)\boldsymbol{B},$$

в которой через $\boldsymbol{A} + \alpha$ обозначен вектор, получаемый из $\boldsymbol{A}$ путем прибавления числа α к каждому компоненту.

В основу функциональной организации конвейера кладется принцип *конвейеризации* [1,2], требующий явного сегментирования арифметико-логического устройства на "специализированные" части. Каждая из таких частей-сегментов должна быть ориентирована на реализацию вполне определенной операции (макро- или микрооперации, в частности) над парой скаляров-операндов (каждый из которых является элементом своего вектора).

Конвейер (pipeline) организуется, в общем случае, как цепочка из элементарных блоков обработки информации (ЭБО$_i$) и памяти (ЭБП$_i$), $i = \overline{1,n}$ (рис.4.1). Каждый из блоков ЭБО$_i$, $i = \overline{1,n}$, осуществляет частичное преобразование $\varphi_i(\boldsymbol{A}, \boldsymbol{B})$ компонентов векторов-операндов:

$$\boldsymbol{A} = \|A_1, A_2, ..., A_i, ..., A_n\|, \quad \boldsymbol{B} = \|B_1, B_2, ..., B_i, ..., B_n\|.$$

Конвейер в целом обеспечивает реализацию достаточно сложного преобразования $\varphi(\boldsymbol{A}, \boldsymbol{B})$, являющегося результатом цепочки преобразований:

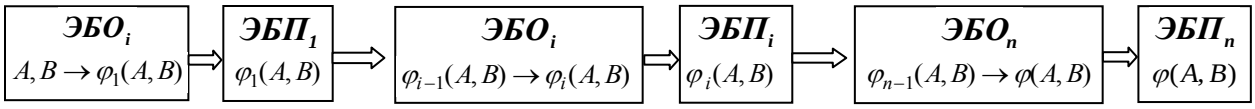
$$\boldsymbol{A}, \boldsymbol{B} \to \varphi_1(\boldsymbol{A}, \boldsymbol{B}) \to \cdots \to \varphi_i(\boldsymbol{A}, \boldsymbol{B}) \to \cdots \to \varphi_n(\boldsymbol{A}, \boldsymbol{B}) = \varphi(\boldsymbol{A}, \boldsymbol{B}).$$

Блоки ЭБП$_i$, $i = \overline{1,n-1}$, и блок ЭБП$_n$ используются для хранения промежуточных результатов $\varphi_i(\boldsymbol{A}, \boldsymbol{B})$ и искомого результата $\varphi(\boldsymbol{A}, \boldsymbol{B})$. Конструктивно блоки ЭБП$_i$, $i = \overline{1,n}$, могут быть объединены в единое целое: в оперативную память либо в векторные регистры.

В простейшем случае элементарные блоки обработки конвейера могут реализовывать отдельные фазы операций (например, арифметических или вычисления элементарных функций), т.е. выполнять микрооперации. Например, при сложении двух вещественных (рациональных) чисел, представленных в форме с плавающей запятой, выполняются следующие микрооперации: сравнение порядков, выравнивание порядков, сложение мантисс, нормализация и т.п. В более общем случае блоки ЭБО$_i$ могут отыскивать промежуточные результаты $\varphi_i(\boldsymbol{A}, \boldsymbol{B})$, $i = \overline{1,n}$, являющиеся, например, или суммой, или разностью, или произведением, или частным для компонентов векторов $\boldsymbol{A}, \boldsymbol{B}$. Как правило, преобразование $\varphi(\boldsymbol{A}, \boldsymbol{B})$ осуществляет одну из арифметических операций над элементами векторов $\boldsymbol{A}, \boldsymbol{B}$.

Элементы векторов подаются в конвейер в дискретные моменты времени и в соответствии с их расположением в векторах. На каждом временном шаге в ЭБО$_1$ конвейера заносится новая пара элементов-операндов текущих векторов $\boldsymbol{A}$ и $\boldsymbol{B}$, а в ЭБО$_i$, $i = \overline{2,n}$, – информация из ЭБП$_{i-1}$ и, в общем случае, извне.

Каноническая структура



Функционирование

$A_1, B_1 \to \varphi_1(A_1, B_1)$				
$A_2, B_2 \to \varphi_1(A_2, B_2)$	…			
…	…			
$A_i, B_i \to \varphi_1(A_i, B_i)$	…	$\varphi_{i-1}(A_1, B_1) \to \varphi_i(A_1, B_1)$		
$A_{i+1}, B_{i+1} \to \varphi_1(A_{i+1}, B_{i+1})$	…	$\varphi_{i-1}(A_2, B_2) \to \varphi_i(A_2, B_2)$	…	
	…	…	…	
$A_n, B_n \to \varphi_1(A_n, B_n)$	…	$\varphi_{i-1}(A_{n-i+1}, B_{n-i+1}) \to$ $\varphi_i(A_{n-i+1}, B_{n-i+1})$	…	$\varphi_{n-1}(A_1, B_1) \to \varphi(A_1, B_1)$
$A_{n+1}, B_{n+1} \to \varphi_1(A_{n+1}, B_{n+1})$	…	$\varphi_{i-1}(A_{n-i+2}, B_{n-i+2}) \to$ $\varphi_i(A_{n-i+2}, B_{n-i+2})$	…	$\varphi_{n-1}(A_2, B_2) \to \varphi(A_2, B_2)$
…	…	…	…	…

Рис.4.1. Конвейерный процессор

ЭБО – элементарный блок обработки информации, **ЭБП** – элементарный блок памяти,
$\boldsymbol{A}, \boldsymbol{B}$ – векторы-операнды; $\varphi_i(\boldsymbol{A}, \boldsymbol{B})$ – частичное преобразование векторов $\boldsymbol{A}, \boldsymbol{B}$

Процесс вычисления $\varphi(\boldsymbol{A}, \boldsymbol{B})$ для пары элементов векторов $\boldsymbol{A}$ и $\boldsymbol{B}$ разделен на n этапов. Все блоки конвейера работают параллельно, но каждый из них реализует свой этап вычислений и обрабатывает свои элементы-операнды в фиксированный момент времени. Очевидно, что время обработки на конвейере конкретных элементов векторов равно суммарному времени их пребывания во всех ЭБО$_i$, $i = \overline{1,n}$. Выдача результатов из "наполненного" конвейера осуществляется через промежутки времени, равные времени выполнения самого медленного этапа. Таким образом, параллелизм в работе блоков конвейера в принципе позволяет достичь производительности, недоступной ЭВМ, базирующимся на модели вычислителя.

Необходимость введения конвейеризации была осознана разработчиками ЭВМ к концу 50-х годов прошлого века. Так, например, в советской ЭВМ М-20 (см. 1.4.3), введенной в эксплуатацию в 1958 г., было реализовано совмещение работы арифметического устройства с выборной очередной команды. Далее, в машине второго поколения ATLAS, разработанной в 1963 г в Манчестерском университете США, выполнение команды было разбито на 4 этапа: выборку команды, вычисление адреса операнда, выборку операнда и выполнение операции. Конвейеризация позволила достичь в ЭВМ ATLAS времени выполнения операции, равного 1,6 мкс (в то время как для последовательной ЭВМ оно было бы равно 6 мкс). Машина БЭСМ-6 (см. 1.4.6), разработанная в 1966 г., характеризуется параллелизмом в работе устройств и конвейерной структурой процессора.