

5.1. КАНОНИЧЕСКАЯ ФУНКЦИОНАЛЬНАЯ СТРУКТУРА МАТРИЧНОГО ПРОЦЕССОРА

Матричный или векторный *процессор* (Array Processor) представляет собой “матрицу” связанных идентичных элементарных процессоров, управляемых одним потоком команд (рис. 5.1). Элементарный процессор (ЭП) включает в себя как арифметико-логическое устройство и память, так и локальный коммутатор. Сеть связей между ЭП (точнее, локальными коммутаторами) позволяет осуществлять обмен данными между любыми процессорами. Поток команд поступает на матрицу ЭП от единого устройства управления (SIMD-архитектура, в каноническом виде).

Архитектура матричного процессора была выбрана в начале 60-х годов 20 века, была обоснована существовавшими экономическими ограничениями и необходимостью обеспечения высокой производительности при решении сложных задач. В самом деле, в то время основная доля стоимости ЭВМ приходилась на схемы устройства управления, а не на схемы арифметико-логического устройства или памяти. Поэтому централизация устройства управления и параллелизм устройств обработки и хранения информации обеспечивали компромисс между стоимостью и производительностью ВС. Однако следует заметить, что с развитием интегральной технологии для производства электронных схем эти экономические соображения стали несущественными при выборе архитектуры вычислительного средства.

Существует широкий спектр сложных научных, технических и экономических задач, которые эффективно решаются на матричных процессорах. При решении сложных задач фактически один и тот же алгоритм параллельно (одновременно) реализуется над многими частями исходного массива данных. Ясно, что перед решением сложной задачи на матричном процессоре требуется предварительная организация данных в векторы или упорядоченные множества. Следовательно, в каждом ЭП размещаются и обрабатываются свои компоненты векторов или свои локальные массивы данных.

Итак, параллелизм в работе элементарных процессоров принципиально позволяет достичь любого уровня быстродействия матричного процессора.

Безусловно матричные процессоры ориентированы на работу в монопрограммном режиме (когда решается только одна задача, представленная в параллельной форме). Однако такие процессоры могут быть использованы и в мультипрограммных режимах (когда решается несколько задач или выполняется несколько параллельных программ с различным числом ветвей, в общем случае). Реализация мультипрограммных режимов в матричном процессоре осуществляется за счет разделения и “времени” и “пространства”. В самом деле, в матричном процессоре имеется единственное устройство управления и множество ЭП, следовательно, в мультипрограммной ситуации должно “делиться” время первого и элементарные процессоры (“пространство”) между программами.

Первой матричной ВС была система SOLOMON (Simultaneous Operation Linked Ordinal MOdular Network). Эта ВС была разработана в Иллинойском университете (University of Illinois) США под руководством Даниеля Л. Слотника (Daniel L. Slotnick).

Планировалось, что система SOLOMON будет иметь матрицу из 32×32 элементарных процессоров, способную выполнять операции над словами с переменной разрядностью, от 1 до 128 разрядов. Каждый ЭП должен был иметь в своем составе арифметико-логическое устройство с последовательной поразрядной обработкой и память емкостью 16 К бит. Все ЭП в данный момент времени могли выполнять только одну и ту же операцию над числами, хранящимися в их ячейках памяти (с одними и теми же адресами). При этом каждый ЭП мог находиться либо в активном состоянии и выполнять команды, поступающие из устройства управления, либо в пассивном состоянии и не реагировать на эти команды.

В качестве устройства управления в системе SOLOMON могла служить серийно выпускаемая ЭВМ. Эта машина должна была иметь память для хранения программ и осуществлять связь с внешними устройствами.

Работы по проекту SOLOMON велись с 1962 г., однако этот проект промышленного воплощения не нашел; в 1963 г. был создан лишь макет ВС размером 3×3 элементарных процессора. Позднее была построена конфигурация ВС размером 10×10 ЭП в фирме Westinghouse Electric Corporation.