

5.4. СЕМЕЙСТВО СИСТЕМ CONNECTION MACHINE

Эволюция архитектуры матричных вычислительных систем и достижения в технологии БИС привели к созданию ВС с массовым параллелизмом, архитектура которых не может быть вписана в какой либо один из канонов. Архитектура данных систем в зависимости от глубины просмотра может быть отнесена к типам MIMD и SIMD одновременно, например, система в целом может иметь тип архитектуры MIMD, а ее основные процессорные компоненты – SIMD, сети связей между элементарными процессорами на различных иерархических уровнях могут быть также различными. К таким ВС относятся модели семейства Connection Machine: CM-1 – CM-5.

Замысел о создании системы Connection Machine возник еще в 1980 г. в фирме MIT и он принадлежит В.Д.Хиллису (W.D.Hillis, MIT AI Laboratory, США). В начале 80-х годов были разработаны архитектура и схемы для создания прототипа этой системы. Актуальность построения системы привела к основанию в 1983 г. фирмы Thinking Machines Corp. (США).

Фирма Thinking Machines Corp. выполнила разработку всех моделей семейства Connection Machine: CM-1 – CM-5. Отметим архитектурные особенности систем семейства CM: тип архитектуры – SIMD (MIMD, системы в целом); массовый параллелизм (Massively Parallel Computer Architecture); максимальное число элементарных процессоров (ЭП) – 65536; однородность и программируемость структуры сети межпроцессорных связей; масштабируемость ВС (возможность создания конфигураций из 16 К, 32 К, 48 К и 64 К элементарных процессоров); быстродействие – до 10^{12} опер./с.

5.4.1. Функциональная структура системы CM

Вычислительные системы семейства CM имеют достаточно сложную функциональную структуру, характеризуются иерархией средств управления процессорами обработки информации (рис. 5.5). В состав системы CM входят: параллельная процессорная система (Parallel Processor Unit); 1-4 коммуникационных процессора (КП 0 – КП 3, Front-ends) с интерфейсом шин (ИШ, Bus interface); матричный коммутатор (МК, Nexus); система ввода/вывода (I/O System); внешние устройства: массовая память данных (ПД, Data Vault), графические дисплеи и др.

Основу системы CM в целом составляет *параллельная процессорная система*, которая может иметь в своем составе до 65536 элементарных процессоров. В этой системе выделяются четыре подсистемы: CM 0 – CM 3. В состав любой из подсистем включаются устройство управления (УУ) и композиция из 16384 элементарных процессоров.

Устройство управления (Sequencer) системы CM – специально спроектированный микрокомпьютер, для реализации виртуальной машины (архитектура которой существенно удобнее для пользователя, чем у реальной физической ВС). Это УУ содержит память микрокоманд емкостью 16 К 96-разрядных слов. В системе CM может содержаться от 1 до 4 устройств управления. На вход устройства управления поступает поток информации “высокого уровня”, именно: команды виртуальной машины и аргументы. Этот поток поступает из матричного коммутатора по синхронному параллельному (32-разрядному) каналу данных. На выходе УУ имеет место поток микрокоманд, которые управляют работой элементарных процессоров и памяти.

Все элементарные процессоры системы CM сгруппированы в *вершины* по 16 ЭП. Каждая вершина конструктивно оформлена как объединение процессорного кристалла и кристалла памяти. Такую вершину в целом называют просто *процессорным кристаллом*. В каждой из подсистем CM 0 – CM 3 имеется 1024 вершины. Вершины взаимодействуют друг с другом через сеть связей, структура которой есть гиперкуб (10-мерный куб).

В качестве *коммуникационного процессора* (Front-end) может быть использована система DEC VAX. Коммуникационные процессоры по сути составляют аппаратно-

программную среду для разработки системного программного обеспечения. Они обеспечивают также взаимодействие с сетью ЭВМ (Network).

Интерфейс шин (Bus Interface) поддерживает 32-разрядный параллельный асинхронный канал между коммуникационными процессорами и матричным коммутатором.

Матричный коммутатор (Nexus) предназначается для организации взаимодействий между коммуникационными процессорами и устройствами управления. Размер коммутатора – 4×4 (4×4 Cross-point switch). Коммутатор реализует механизм разделения, который позволяет в пределах системы СМ конфигурировать до 4 подсистем, работающих под управлением своего коммуникационного процессора. Это позволяет применять различные подсистемы для решения задач различных пользователей или, например, выделять одну из них для диагностирования и восстановления работоспособности, а остальные использовать для решения задач. В случае, если к одному коммуникационному процессору подсоединяется более одного устройства управления, осуществляется их синхронизация (при помощи генератора синхроимпульсов коммутатора).

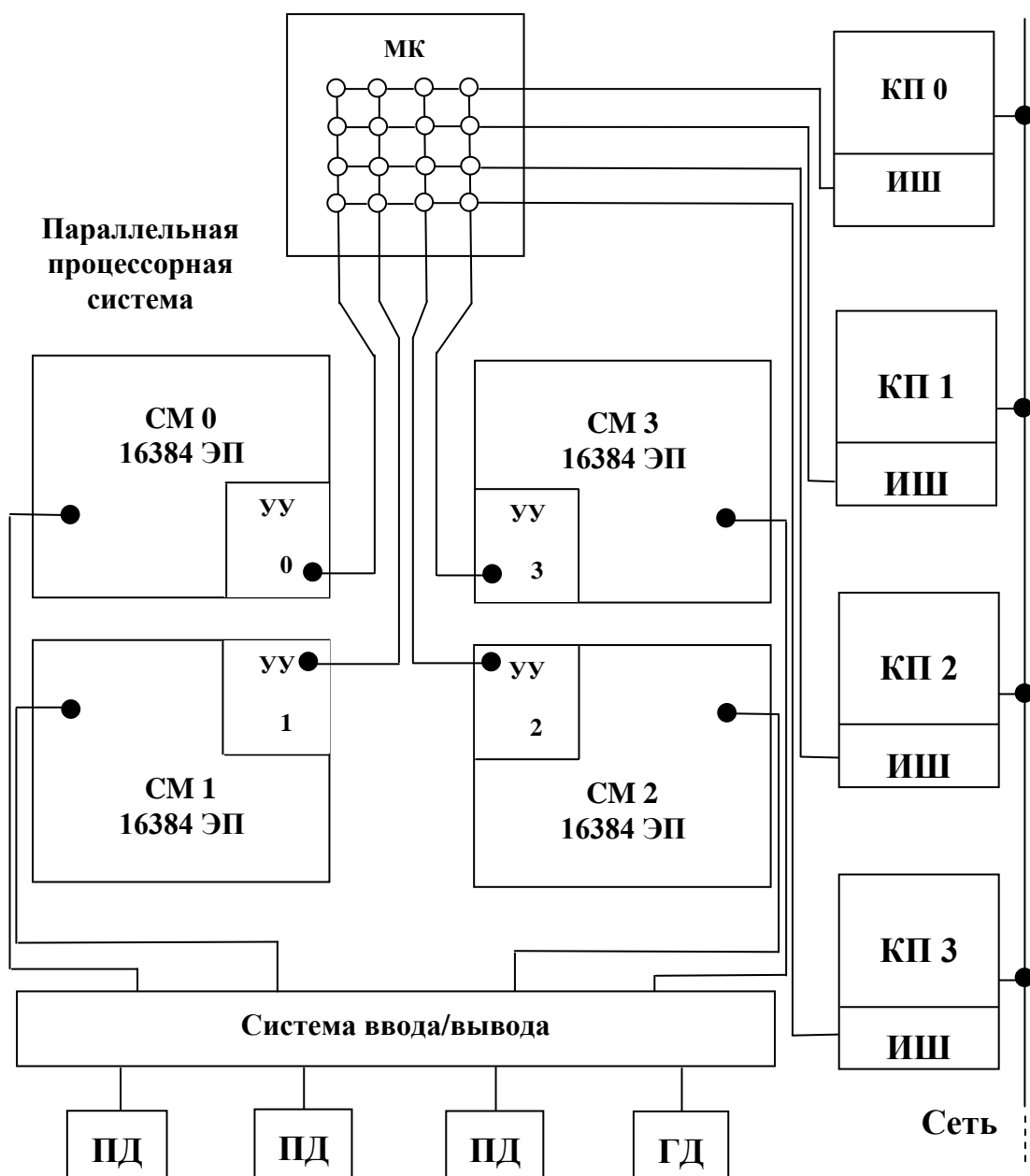


Рис. 5.5. Функциональная структура системы СМ

5.4.2. Элементарные процессоры системы СМ

Элементарный процессор системы СМ – это основной функциональный элемент параллельной процессорной системы (рис.5.5). В состав базовой конфигурации любого ЭП (рис.5.6) входят:

- арифметико-логическое устройство (АЛУ);
- битноадресуемая локальная память (ЛП);
- одноразрядные регистры признаков (флагов):
- интерфейс маршрутизатора;
- интерфейс сети межпроцессорных связей.

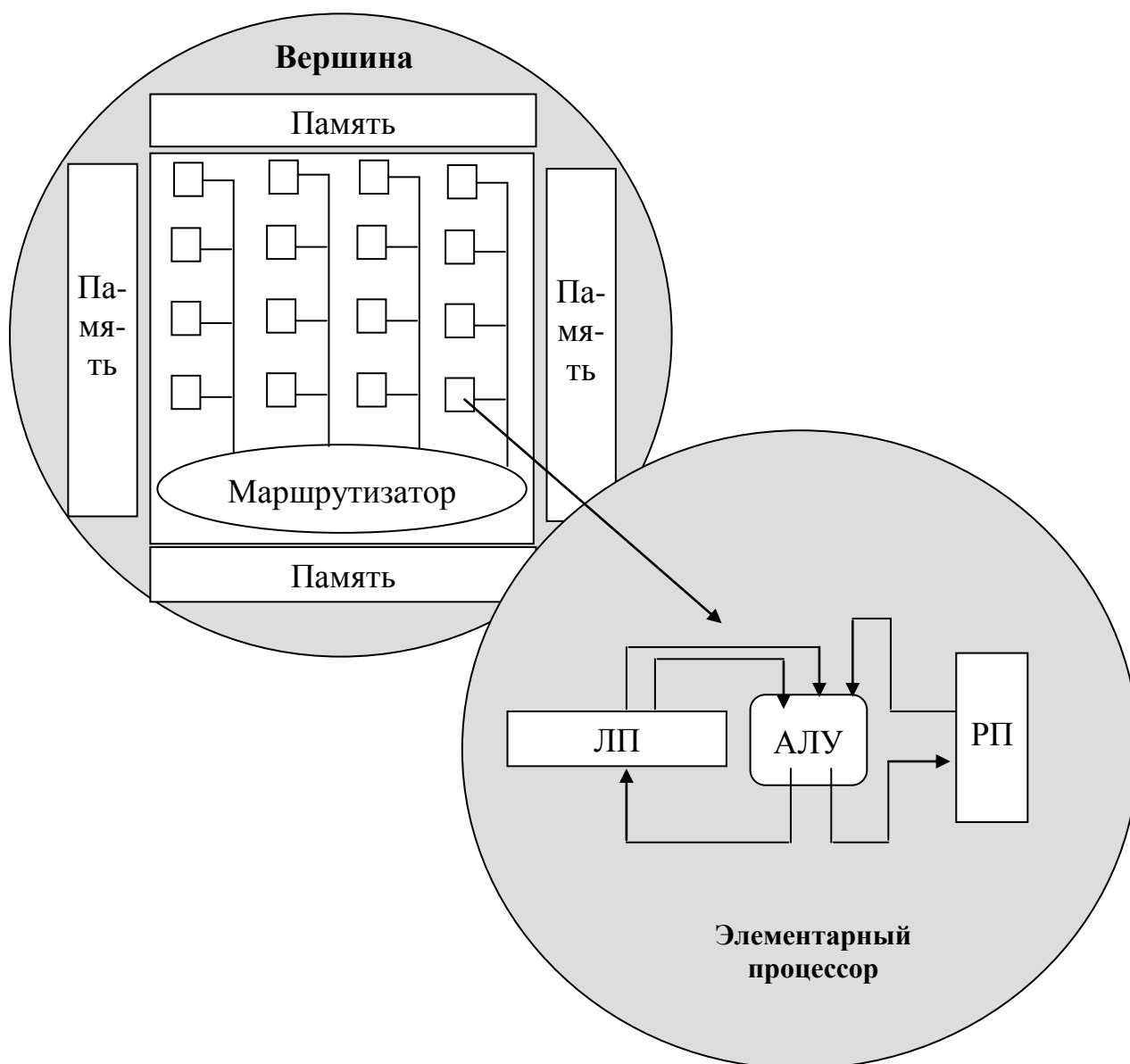


Рис.5.6. Функциональная структура вершины и элементарный процессор системы Connection Machine, АЛУ – арифметико-логическое устройство, ЛП – локальная память, РП – регистры признаков

Элементарный процессор системы CM не является конструктивно оформленным элементом. В качестве конструктивной (да и функциональной) единицы выступает вершина (рис.5.6). При реализации вершины используются два типа кристаллов. Первый из них – это оригинальный специально спроектированный (заказной) кристалл (proprietary custom chip). Этот кристалл называют процессорным, он содержит арифметико-логические устройства, регистры признаков и коммуникационный интерфейс для 16 элементарных процессоров. Второй кристалл – это коммерческая статическая память с произвольным доступом и с защитой по четности. Итак, в полной конфигурации системы CM (из 64 К элементарных процессоров) содержится 4096 процессорных кристаллов (вершин) и имеется память с произвольной выборкой емкостью $65536 \cdot \nu$ бит, где ν – емкость локальной памяти одного ЭП.

Арифметико-логическое устройство ЭП имеет три входа и два выхода и включает в себя логические элементы, одноразрядные регистры-защелки (latches) и интерфейс памяти (рис.5.6). Опишем цикл функционирования АЛУ. Вначале осуществляется чтение двух одноразрядных операндов из памяти и одного бита данных из регистра признака. Затем логические элементы выполняют операцию над этими тремя входными битами и выдают два одноразрядных результата. Наконец, один из двух результатов запоминается в памяти, а другой – в регистре признака. Выполнение операции в целом зависит от значения признака: если он равен нулю, то результаты в данном ЭП не запоминаются.

Логические элементы АЛУ способны вычислять значения любых двух булевых функций от трех переменных. Не смотря на свою простоту АЛУ способно выполнять все операции из набора команд виртуальной машины. Арифметические операции выполняются последовательно, побитно. Пусть τ – цикл АЛУ, он включает в себя время выполнения операции над битами информации плюс время декодирования команды и накладные расходы. Тогда время сложения L -разрядных операндов потребует $L\tau$ единиц времени, а номинальное быстродействие системы CM из 64 К элементарных процессоров составит $65536/L\tau$ операций в единицу времени. Например, для CM-1 цикл $\tau = 0,75$ мкс, тогда быстродействие при выполнении операций сложения 32-разрядных чисел оценивается величиной $2 \cdot 10^9$ опер./с.

Элементарный процессор системы CM – это RISC-компьютер (RISC – Reduced Instruction Set Computer, компьютер с упрощенным набором команд). Каждый цикл АЛУ подразделяется на подциклы. В каждом подцикле элементарные процессоры выполняют команду низкого уровня, называемую нанокондандой (nanoinstruction), поступающую из устройства управления, в то время как памяти могут выполнять одну операцию чтения или записи. Основной цикл АЛУ при выполнении целочисленной операции сложения двух операндов состоит из трех подциклов, соответствующих следующим нанокондандам: LoadA – чтение из памяти операнда A, LoadB – чтение из памяти операнда B и Store – запоминание результата. Имеются также наноконданды управления маршрутизатором и межпроцессорной сетью связей (NEWS grid), а также команды, реализующие функции самодиагностики (в модели CM-1 используется двумерная сеть, это и предопределило аббревиатуру NEWS: North – East – West – South).

5.4.3. Сеть межпроцессорных связей системы CM

Вычислительные системы семейства Connection Machine имеют развитую программно настраиваемую сеть межпроцессорных связей. В зависимости от типа и структуры решаемой задачи программируются межпроцессорные взаимодействия в системе CM. Чтобы выразить отношения между элементарными процессорами, разработчики алгоритмов обычно используют технику структурирования данных. Например, в системах распознавания образов для представления пикселей образов обычно

используются двумерные решетки из элементарных процессоров (рис.5.7). Однако на поздней стадии обработки для представления более абстрактных отношений между объектами и их частями могут применяться структуры в виде дерева или реляционные графы (графы отношений).

Следует заметить, что в последовательной ЭВМ, имеющей память с произвольной выборкой, для создания сложных структур данных используется указатель элементов памяти. В архитектурах ВС с массовым параллелизмом индивидуальные элементы данных назначаются на индивидуальные ЭП, а отношения между элементами очень больших структур данных реализуются через межпроцессорные коммуникации.

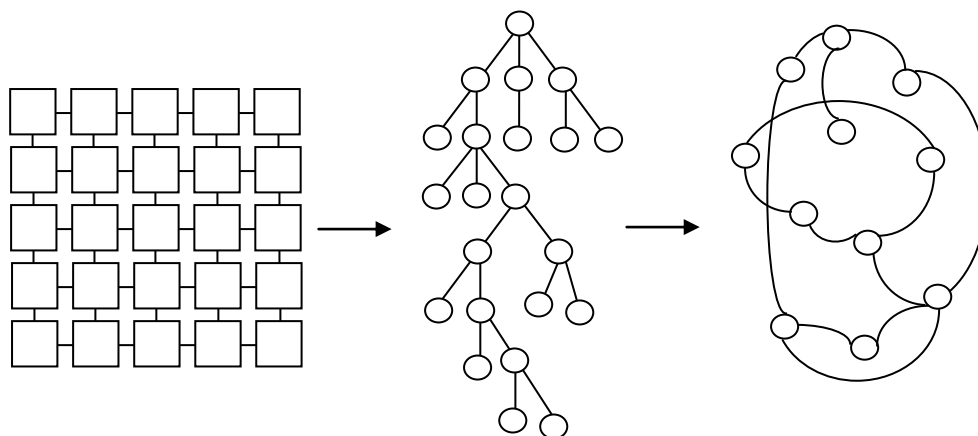


Рис. 5.7. Программирование структуры сети межпроцессорных связей системы СМ

В сети межпроцессорных связей системы СМ поддерживаются ниже следующие коммуникационные механизмы.

Широковещательная связь (Broadcast Communication) обеспечивает безотлагательный прием всеми одновременно элементарными процессорами данных, поступающих от устройства управления или коммуникационного процессора.

Связь “обобщенное ИЛИ” (Global OR) осуществляет реализацию логического ИЛИ над выходными разрядами переноса АЛУ всех элементарных процессоров, что делает возможным быстро обнаруживать необычные условия либо условия окончания вычислений.

Гиперкубическая связь (Hypercube Communication) является средой для маршрутизатора и многочисленных параллельных примитивов, поддерживаемых моделью виртуальной машины. Топология сети межпроцессорных связей системы СМ представляется гиперкубом, т.е. булевым n -кубом. Например, в конфигурации ВС из 16384 ЭП используется 10-куб, объединяющий 1024 вершины (16-процессорных кристалла).

Маршрутизатор (Router) 16-процессорного кристалла (рис.5.6) непосредственно осуществляет пересылку пакетов сообщений в элементарные процессоры. Адреса процессоров содержатся в указателе (Pointer), указатель следует вместе с пакетом сообщений. Контроллер маршрутизатора (реализованный в процессорном кристалле) использует гиперкуб для передачи данных. Эти аппаратные средства обеспечивают частичное совмещение коммутации сообщений с решениями о маршрутизации, с буферизацией и объединением сообщений, направляемых по некоторому адресу.

Межпроцессорные связи (NEWS grid) – декартова решетка, обеспечивающая прямой доступ из данного ЭП к ближайшим соседям в процессорном кристалле. Поскольку при использовании этой сети связей все процессоры передают информацию в одном и том же направлении (на “север”, “восток”, “запад” или “юг”), а адреса задаются

неявно и никаких коллизий не происходит, то она заметно быстрее (примерно в 6 раз), чем связь при использовании маршрутизатора (для простых регулярных сообщений).

5.4.4. Модель виртуальной машины системы СМ

Пользователю системы СМ предоставляется удобный “сервис” – виртуальная машина. Архитектура этой машины весьма близка к архитектуре физической системы Connection Machine, но имеются все же два существенных расширения. Это то, что набор параллельных команд виртуальной машины (названный Paris – Парис) много богаче реального, и то, что имеется абстракция виртуального процессора.

Набор команд Paris перекрывает достаточно широкий диапазон от простых логических и арифметических операций до высокоуровневых операций, таких как APL-операции (APL – A Programming Language), сортировка и коммуникационные операции. Функции интерфейса Paris (между коммуникационным процессором и остальной частью системы СМ) сводятся к формированию потока кодов операций и аргументов. Аргументы – это, как правило, начальный адрес и количество бит (разрядность операнда). В качестве аргумента могут служить непосредственно данные или информация о широкополосном приеме. Большая часть набора Paris реализована в аппаратуре устройств управления. В УУ осуществляется синтаксический анализ потока кодов операций и аргументов и его преобразование в соответствующую последовательность нанокоманд для элементарных процессоров. Поскольку Paris определяет набор команд виртуальной машины, то мы можем использовать те же самые имена и для языка ассемблера системы СМ.

Виртуальный процессор. Во многих областях параллельной обработки данных часто требуются специфические процессоры, которые заметно отличаются от физических ЭП данной системы. Программное обеспечение системы СМ предоставляет механизм виртуального процессора, он поддержан на уровне Paris и понятен пользователю. При инициализировании системы СМ указывается число виртуальных процессоров, которое требуется в конкретном применении ВС. В случае, если это число превышает количество физических ЭП, то локальная память каждого ЭП расщепляется на области, как это требуется, а время процессоров автоматически делится между этими областями.

Например, если необходимо обработать 1 М порций данных, то естественно требуется $V = 2^{20}$ виртуальных процессоров. Пусть реальное число физических ЭП равно $P = 2^{16}$ и каждый процессор имеет локальную память емкостью 2^m бит (для модели СМ-1 $m = 12$, а для СМ-2 $m = 16$). Тогда каждый физический ЭП будет поддерживать $V/P = 16$ виртуальных процессоров.

Отношение $V/P = k$ называют коэффициентом виртуального процессора. В нашем примере каждый виртуальный процессор имел бы память емкостью $M/k = 2^{m-4}$ бит и обрабатывал бы коды с быстродействием, которое составляло бы всего $1/k = 1/16$ от быстродействия физического ЭП. Фактически виртуальные процессоры часто превышают эту скорость вычислений; расходы времени на декодирование команд в УУ могут быть “погашены” за счет избытка числа виртуальных процессоров.

5.4.5. Программное обеспечение системы СМ

Основу системного программного обеспечения составляет операционная система (Operating System), являющаяся штатной операционной средой (либо UNIX либо LISP) коммуникационных процессоров с небольшим расширением. Пользователям предоставляется возможность применять обычные языки и конструкции программирования со всем программным инструментарием коммуникационных

процессоров. Таким образом, пользователи могут без особого труда разработать программы, рассчитанные на эксплуатацию всей вычислительной мощности аппаратуры системы Connection Machine.

Тем не менее следует заметить, что стандартные языки программирования все же имеют некоторые расширения, поддерживающие параллельные конструкции данных. Однако эти расширения не требуют изучения какого-либо нового стиля программирования.

Язык FORTRAN системы Connection Machine использует расширения (Array Extentions) для работы с векторами, матрицами и массивами данных в стандарте FORTRAN 8x (предложенном American National Standards Institute Technical Committee). Эти расширения естественно отображаются в основном параллельном оборудовании системы CM.

Языки *Lisp и CM-Lisp являются параллельными диалектами обычного языка Lisp. Язык *Lisp дает программистам возможность высококачественно управлять аппаратурой системы CM при сохранении своей гибкости. CM-Lisp является языком высокого уровня, он “пристраивает” небольшие синтаксические дополнения в интерфейс языка и тем самым создает мощный язык параллельного программирования.

Язык C* является параллельной версией C. Программы на C* могут быть прочитаны и записаны также как последовательные C-программы. Расширения незначительные и легко воспринимаются.

Язык ассемблера (точнее Paris) системы CM является объектным (выходным) языком компиляторов языков высокого уровня. Набор команд Paris логически расширяет набор команд коммуникационного процессора и маскирует физическую реализацию системы CM.

5.4.6. Модели семейства Connection Machine

Модель CM-1. Первая модель системы Connection Machine была спроектирована в фирме Thinking Machines в течении 1983 г. и первой половины 1984 г. Прототип модели CM-1 из 16 К процессоров был построен к концу 1984 г. при финансовой поддержке Агентства по перспективному планированию исследований в области обороны США (Defense Advanced Research Projects Agency). Демонстрация возможностей прототипа CM-1 была осуществлена в мае 1985 г. Полная 65536-процессорная конфигурация CM-1 была собрана и успешно продемонстрирована в ноябре 1985 г. Коммерческая презентация CM-1 была осуществлена фирмой в апреле 1986 г., а первые поставки произведены летом 1986 г. (в фирмы MIT и Perkin-Elmer).

Модель CM-1 включала следующие компоненты:

- коммуникационные процессоры – 1-4,
- устройства управления – 1-4,
- элементарные процессоры – до 64 К,
- сеть межпроцессорных связей.

Система ввода/вывода информации в модели CM-1 реализована не была.

Элементарный процессор модели CM-1 – это базовая конфигурация, включающая:

- арифметико-логическое устройство;
- локальную память емкостью 4 К бит;
- 8 регистров признаков (флагов);
- интерфейс маршрутизатора;
- двумерный интерфейс сети межпроцессорных связей.

Конфигурация ЭП предопределяет двумерную сеть связей (NEWS griol) в пределах 16-процессорного кристалла (вершины гиперкубической структуры).

Модель СМ-1, включавшая 64 К элементарных процессоров имела быстродействие 2000 MIPS ($2000 \cdot 10^6$ операций в секунду над 32-разрядными целыми числами) и обладала оперативной памятью емкостью 32 К байт.

Модель СМ-2. Модель СМ-2 эта развитая версия СМ-1, в ней нашли отражение опыт эксплуатации первой модели и следующие цели:

- обеспечение совместимости,
- увеличение производительности и емкости памяти,
- повышение общей надежности,
- упрощение производства,
- подключение высокоскоростной системы ввода/вывода (для внешней памяти и дисплеев).

Процессорный кристалл модели СМ-2 был спроектирован в начале 1986 г., а первый прототип СМ-2 был построен уже в конце 1986 г. Коммерческое представление модели СМ-2 осуществлено в апреле 1987 г., а к осени 1987 г. была произведена поставка покупателям около дюжины моделей. Первая массовая память (дисковая память с массовым параллелизмом, названная Data Vault) была поставлена в конце 1987 г.

Архитектуры элементарных процессоров моделей СМ-1 и СМ-2 совместимы, отметим основные особенности ЭП модели СМ-2:

- емкость бит-адресуемой памяти – 64 К бит (вместо 4 К бит);
- 4 одно-разрядных регистра признаков (вместо 8);
- акселератор для операций с плавающей запятой, необязательный (Accelerator – ускоритель операций);
- универсальный интерфейс для поддержки n -мерных решеток (вместо двумерных, NEWS-решеток);
- интерфейс ввода/вывода;
- развитые схемы обнаружения ошибок.

Для реализации элементарных процессоров модели СМ-2 используется четыре (а не два) типа кристаллов. Один из кристаллов – это специально спроектированный (заказной) процессорный кристалл, содержащий АЛУ, регистры признаков, интерфейс маршрутизатора, интерфейс n -мерных решеток и интерфейс ввода/вывода для 16 элементарных процессоров и часть контроллера для организации гиперкубической сети. Второй кристалл – это коммерческая динамическая память с произвольной выборкой, в которой реализуется коррекция однобитных ошибок и обнаружение двухбитных ошибок. Еще два заказных кристалла используются для формирования акселератора, причем один из них реализует операции с плавающей запятой, а другой – функции интерфейса. Композиция из последних двух кристаллов рассчитана на подключение 32 элементарных процессоров. Итак, в полной 65536-процессорной конфигурации системы СМ-2 содержится 4096 процессорных кристаллов, 2048 кристаллов для операций с плавающей запятой, 2048 кристаллов для интерфейса с плавающей запятой и оперативная память емкостью 512 М байт.

Сеть межпроцессорных связей в модели СМ-2 почти такая же как в СМ-1, однако в ней имеются две особенности. Во-первых, маршрутизатор процессорного кристалла СМ-2 более производительный и надежный и поддерживает диагностику. Во-вторых, в пределах процессорного кристалла (вершины гиперкуба) допускается формирование n -мерных решеток из элементарных процессоров. Последнее позволяет пользователям в зависимости от решаемой задачи применять многомерные решетки для межпроцессорных связей.

Система ввода/вывода модели СМ-2 содержит 8 контроллеров, которые обеспечивают скорость обмена информацией между параллельной процессорной системой и внешними устройствами, равную 320 М байт в секунду. Полоса пропускания шины ввода/вывода – 80 бит (64 бита – для данных, 8 бит – для контроля по четкости, 8 бит – для управления).

Внешняя массовая память (Data Vault) состоит из 8 устройств. Каждое устройство способно хранить от 5 до 10 М байт данных и обеспечивает обмен информацией со скоростью 40 М байт в секунду. Таким образом, массовая память обеспечивает суммарную емкость 8 G байт и скорость обмена информацией 320 М байт в секунду.

Быстродействие модели СМ-2 существенно выше, чем у СМ-1. Так при выполнении логических и арифметических операций достигается *быстродействие 2500 MIPS* ($2500 \cdot 10^6$ операций в секунду над 32-разрядными целыми числами). *Акселератор СМ-2 повышает быстродействие системы более чем 20 раз* при выполнении операций с плавающей запятой. Емкость оперативной памяти системы СМ – 512 М байт, а массовой внешней памяти – 8 G байт.

Модель СМ-5. Архитектурное развитие семейства СМ и возможность применения все более совершенных микропроцессорных БИС привели к созданию высокопроизводительной модели СМ-5. *Производительность конфигурации СМ-5, состоящей из 16384 элементарных процессоров* оценивается величиной 10^{12} *FLOPS* (10^{12} операций с плавающей запятой).