

ГЛАВА 1

ОСОБЕННОСТИ ОРГАНИЗАЦИИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

ПРЕДИСЛОВИЕ

Параллельная ВС — это набор процессоров, способных совместно решать некоторую вычислительную задачу [82]. Данное определение достаточно широко и включает в себя: параллельные супер ЭВМ, которые содержат сотни и тысячи процессоров, сети рабочих станций, мультипроцессорные рабочие станции и встроенные системы. Параллельные ЭВМ интересны тем, что в них заложены огромные возможности по концентрации вычислительных ресурсов, таких как процессоры, память, либо пропускная способность устройств ввода/вывода для решения важнейших вычислительных задач.

Параллелизм иногда рассматривается как редкая и экзотическая область вычислительной техники, интересная, но не имеющая практического применения и не используемая обычными программистами. Изучение тенденций развития программных приложений, архитектуры ЭВМ и вычислительных сетей показывает, что параллелизм в настоящее время является вездесущим, а параллельное программирование становится основным для разработчиков программного обеспечения для современных ВС.

По мере того как компьютеры становятся все более быстрыми, можно справедливо предположить, что в конце концов они станут "достаточно быстрыми" для практических приложений и необходимость в увеличении мощности компьютеров уйдет в прошлое.

Однако, история показывает, что определенная технология, удовлетворяющая некоторое существующее программное обеспечение, способствует появлению новых программных продуктов, которые требуют развития новых технологий. В качестве интересной иллюстрации этого феномена можно привести отчет, составленный для правительства Великобритании в 1940 году, из которого следует, что по крайней мере две или возможно даже три ЭВМ способны удовлетворить возникшие потребности, связанные с вычислениями. В то время компьютеры использовались преимущественно для расчетов баллистических таблиц, так что авторы этого отчета не учитывали других научных, инженерных и коммерческих программных приложений.

Традиционно, развитие компьютеров высокого класса мотивировалось необходимостью численного моделирования сложных систем, таких как: погода, климат, механические устройства, электрические цепи, производственные процессы и химические реакции. Однако, в настоящее время наиболее значительными факторами, управляющими развитием высокопроизводительных ЭВМ, являются коммерческие приложения, которые требуют от компьютера способности оперировать большими объемами данных. Эти приложения включают в себя видеоконференции, компьютерную медицинскую диагностику, параллельные базы

данных для поддержки решений, а также усовершенствованную графику, виртуальную реальность и многое другое. Интеграция параллельных вычислений и технологий мультимедиа ведет высокопроизводительные компьютерные сети к развитию *видео услуг*, т. е. проектируются компьютеры, обслуживающие сотни и тысячи одновременных запросов для формирования видеоизображений в режиме реального времени. Причем, каждый видеопоток может содержать в себе как передачу данных со скоростью нескольких мегабайт в секунду, так и огромное количество операций по кодированию и декодированию данных.

Несмотря на то, что коммерческие приложения могут определять архитектуру большинства будущих параллельных компьютеров, тем не менее традиционные научные программные приложения останутся важными для пользователей параллельной вычислительной технологии. Дело в том, что результаты, полученные чисто теоретическими методами, в отдельных приложениях требуют подтверждения практикой. Вместе с тем, проведение натурных экспериментов становится все более дорогим и непрактичным, поэтому компьютерное изучение сложных систем становится все более важным. Возрастающая требовательность исследователей к точности проводимых вычислений увеличивает вычислительные затраты таких расчетов и определяют необходимость повышения мощности и/или "разрешающей способности" вычислительной системы. Однако увеличивающиеся возможности ВС позволяют пользователям использовать их для более глубоких исследований, требующих более сложных расчетов. Этот подход имеет практически неудовлетворяемые требования к большой вычислительной мощности. Кроме того, ВС характеризуются требованиями к большим объемам памяти и повышенными требованиями к устройствам ввода/вывода и отображения информации. К примеру, моделирование десятилетнего изменения климата Земли, использующее современные модели, может включать в себя до 10^{16} операций с плавающей точкой — на это требуется десять дней при скорости выполнения 10^{10} операций с плавающей точкой в секунду. Кроме того, решение задач такой сложности может легко генерировать сотни и даже большее гигабайт данных.

В общем случае, необходимость в высокоскоростных компьютерах определяется как большими объемами данных в коммерческих приложениях, так и огромным количеством вычислений в научных и инженерных приложениях. И по мере того, как инженерные приложения используют все возрастающее количество данных, а в коммерческих приложениях появляются сложные расчеты, происходит все большее объединение запросов этих двух областей прикладного программирования.

С 1945 года и до настоящего времени производительность высокоскоростных компьютеров возрастала по экспоненциальному закону с коэффициентом роста равным 10 каждые 10 лет. В то время, как первые ЭВМ выполняли до нескольких десятков операций с плавающей точкой в секунду, производительность параллельных компьютеров середины 90-х годов достигает десятков миллиардов операций в секунду (рис. 1.1).

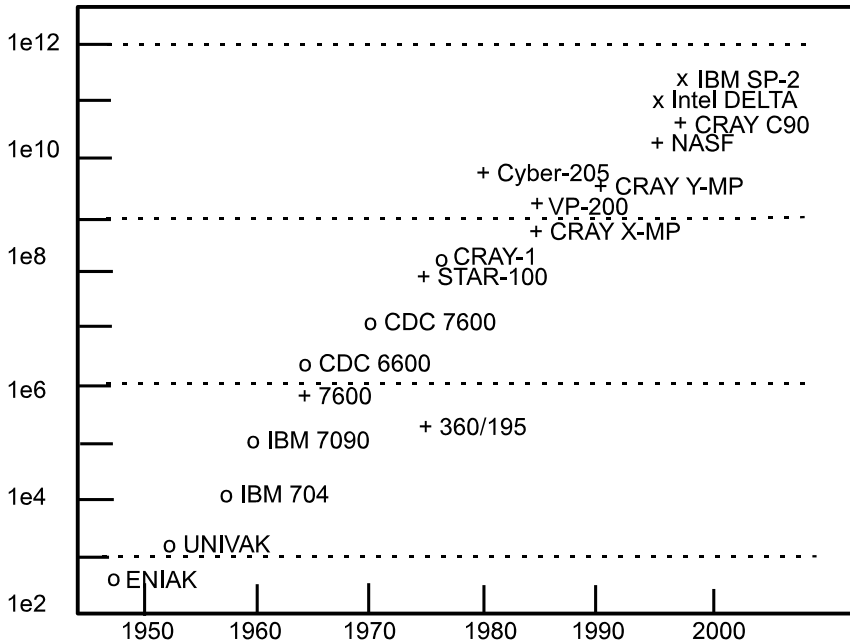


Рис.1.1 (по вертикали указано количество операций с плавающей точкой в (секунду) *Высшие показатели производительности некоторых высокоскоростных суперкомпьютеров, 1945 — 1995. Экспоненциальный рост кривой несколько спадает в 1980-х, однако, он снова возрастает по мере того, как появляются параллельные суперкомпьютеры. Символ "o" означает однопроцессорные системы, "+" — параллельные векторные компьютеры с 4-16 процессорами и символ "x" — означает мощные параллельные ЭВМ с сотнями или тысячами процессоров.*

Аналогичные тенденции наблюдаются и в компьютерах более низкого класса, принадлежащих к разным поколениям: калькуляторах, персональных компьютерах и рабочих станциях. И мало вероятно, что этот рост не будет продолжаться в дальнейшем. Тем не менее, архитектура компьютеров, поддерживающая данный рост, радикально изменяется от последовательной к параллельной.

Производительность компьютеров зависит от времени, необходимого для выполнения основных операций или "**тактового цикла**" процессора, т. е. времени, необходимого для выполнения элементарных операций. Однако, времена тактовых циклов уменьшаются медленно и ограничены величинами физического характера, такими как скорость света (рис 1.2). Можно с уверенностью сказать, что в будущем увеличение производительности вычислений в ВС мало будет зависеть от скорости самого процессора.

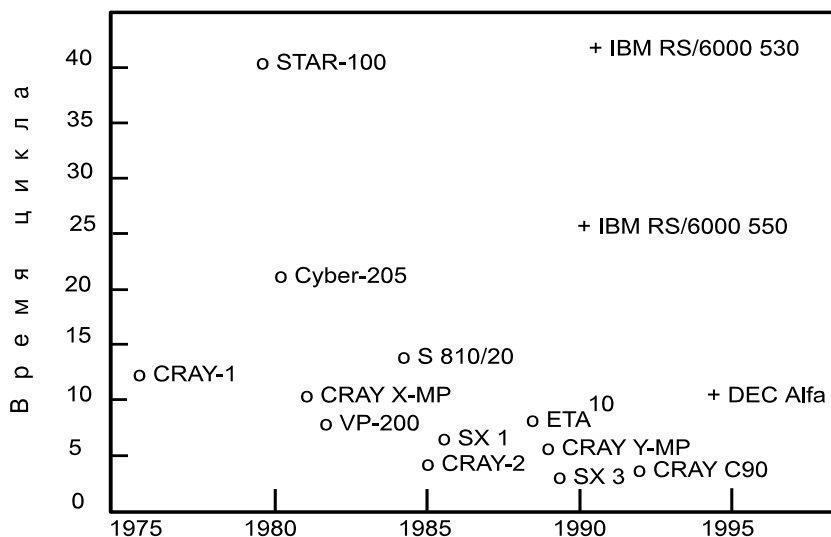


Рис 1.2 (время цикла дано в наносекундах)

Тактовые циклы векторных суперкомпьютеров (обозначены символом "o") уменьшились за шестнадцать лет в три раза, начиная от CRAY-1 (12.5 наносекунд) и заканчивая C90 (4.0). RISC микропроцессоры (обозначены символом "+") быстрее приближались к подобной производительности. Обе архитектуры похоже уже близки к физическим пределам.

Для того, чтобы обойти эти ограничения используется внутренний параллелизм в самом чипе (кристалле). Применяются и другие различные подходы, включая использование конвейерного режима (различные стадии нескольких команд выполняются одновременно) и многофункциональных устройств (несколько умножителей, сумматоров, контролируемых одним потоком команд). Все больше и больше внедряются "мультикомпьютеры", каждый из которых имеет свой процессор, память и ассоциативную взаимосвязанную логику. Этот подход перспективен благодаря развитию VLSI технологии и позволяет уменьшать количество технологических компонентов, встраиваемых в компьютер. Т.к. стоимость компьютера (очень приблизительно) пропорциональна количеству содержащихся в нем компонент, то увеличение уровня их интеграции ведет к увеличению количества процессоров, входящих в компьютер, при той же относительной цене.

1.1 МОДЕЛИ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Общепринято рассматривать отдельно различные классы моделей для последовательных и параллельных ВС: машинные, архитектурные, вычислительные и программные [12,16,30,52,144]. Эти классы различаются не только по назначению, но и также по уровню абстракции.

Для организации и планирования параллельных вычислений в многопроцессорных вычислительных системах, комплексах и сетях целесообразно рассматривать два типа моделей:

- * машинная модель, которая определяется архитектурой компонентов ВС;
- * программная модель вычислений, включающая различные компоненты для выполнения и отображения алгоритма.

В данной работе рассматриваются модели ВС, которые уже внедрены или, возможно, будут внедрены в ближайшем будущем. Все обобщения, сведенные в модели параллельных вычислительных систем, базируются на концепциях универсальной алгебры представленной в понятиях абстрактной алгебры [90]:

Абстрактная алгебра использует двойку (E, F) , где E — не пустое множество элементов, а F — множество (внутреннее или внешнее) операций, определенных на E .

Собственно говоря, ВС выполняет то, что определено в абстрактной алгебре: использование или обращение (автоматическое или нет) исполняемых операций (команд) к данным. Единственное различие лежит в понятии исполнительной среды: в случае абстрактной алгебры обращение к операциям, описанным множеством F , и их выполнение осуществляет пользователь; для компьютера исполнительной средой является устройство управления, а его поведение определяется программой.

Оговоренные понятия приводят к нижеследующему обобщенному описанию модели компьютера:

$$\text{компьютер} = \text{данные} + \text{операции} \quad (1)$$

Это выражение является исходным для последующих заключений.

* *Общие характеристики моделей вычислительных систем*

Анализ уравнения (1) приводит к выводу, что любая модель ВС должна иметь такие характеристики, которые бы допускали возможность применения концепции универсальной алгебры, т. е. способ представления данных (и доступа к ним), и выполнение операций с этими данными.

Кроме того, любая модель должна определять:

- * существование двух синергических компонент:
 - памяти, как физической поддержки представления данных;

— аппаратной части для управления выполнением операций (используя достоверные данные);

- * возможность произвольной, прямой (параллельной или последовательной) выборки данных;
- * число возможных параллельно выполняемых операций или одновременно выполняемых операций с памятью;
- * обмен данными между синергическими компонентами (памятью и аппаратной частью) осуществляется как выполнение команд в логическом устройстве.

Параллельность доступа к памяти или выполнения команд — это критерии, которые переводят компьютеры в разряд параллельных или последовательных машин.

Модель ВС не определяет конкретный способ ее физической реализации.

Например, реальный доступ к памяти ВС может быть реализован следующими способами:

- * случайный или прямой, когда исполнительная часть имеет непосредственную физическую связь с памятью (bus), что дает возможность обращаться к любой ячейке памяти;
- * через сетевую взаимосвязь (network), которая является обобщением общей шины памяти.

Требования к выполнению программы также заключаются в том, что программа и данные должны находиться в памяти, какая бы ни была физическая реализация доступа к ней (реализация фон Неймана или Гарварда). Например, системы I80x86 имеют память с сегментной организацией, в которой одновременно хранятся данные и программы, а в вычислительных машинах с массовым параллелизмом, таких как Connection Machine [96], Sphinx [130], или MasPar MP-1 [122], данные передаются параллельно и располагаются в рабочих ОЗУ исполнительных устройств, а команды — в памяти центральной (host) вычислительной машины.

1.2. МОДЕЛЬ С ПОСЛЕДОВАТЕЛЬНЫМ ДОСТУПОМ К ПАМЯТИ (RAM МОДЕЛЬ)

Некоторые модели ВС [97]) определены для последовательных вычислительных машин. Модель Тьюринга (Turing machine model) или конечный автомат (который по сути является частным случаем машины Тьюринга) отличается ограниченностью характеристик случайного доступа к памяти.

Память с RAM доступом [67] хорошо подходит для проектирования и организации последовательных ВС и хорошо исследована. Эта модель имеет исполнительную (рабочую) часть, которая взаимодействует с основной памятью. ВС с RAM имеют SISD архитектуру (рис.1.4.).



Рис. 1.4. Структура ВС SISD организации с RAM памятью

Модель с RAM организацией памяти позволяет выполнять только одну команду в каждый момент времени, независимо от того, какой окажется данная команда (ссылкой к памяти или арифметической операцией).

Операции, выполняемые на одном исполнительном устройстве, следуют в жесткой последовательности, не допускающей параллелизма, даже если архитектурные усовершенствования допускают конвейерное выполнение некоторых из операций (не все операции требуют для своего выполнения всех внутренних устройств) или их параллельно-подобное выполнение (например, в случае с DMA, Direct Memory Access — прямой доступ к памяти — способ обмена данными между адаптером и памятью без участия процессора, что может заметно снизить нагрузку на процессор и повысить общую производительность системы). Для повышения эффективности работы ЭВМ также используется организация просмотра команд вперед или наличие нескольких исполнительных блоков, а также использование многоуровневой кэш памяти или внутреннее процессорное буферирование запросов записи.

Программа и данные хранятся в памяти с прямым или случайным доступом. Практически, каждый одноадресный компьютер с организацией фон Неймана — это RAM машина. Анализ сложности алгоритмов для RAM машин сводится к анализу времени выполнения отдельных команд или программ, работающих в многопрограммном режиме, при реализации истинного или кажущегося совмещения.

1.3 МОДЕЛИ ПАРАЛЛЕЛЬНЫХ МАШИН

Параллельные компьютеры, так же как и последовательные состоят из двух функциональных компонент: *исполнительной (процессора)* и *памяти*. Различия между ними состоит только в количестве используемых компонент (рис.1.5).

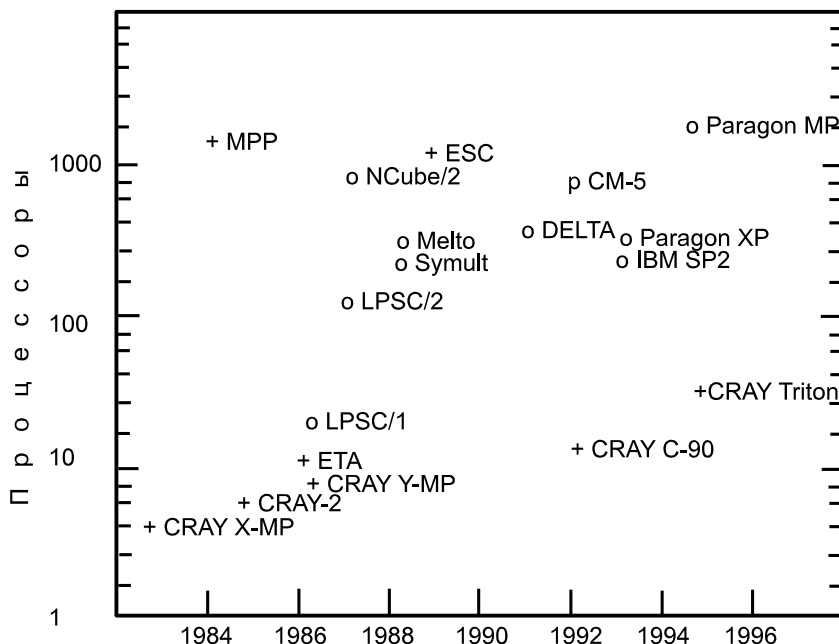


Рис. 1.5. Число процессоров в мощных параллельных компьютерах ("o") и в векторных мультипроцессорах ("+")

Особенности архитектуры параллельных ВС (рис.1.6) и организации параллельных вычислений позволяют разделить их на несколько классов: SIMD, MISD, MIMD (классификация Флина (Flynn's classification) 1966) [12, 97, 130].

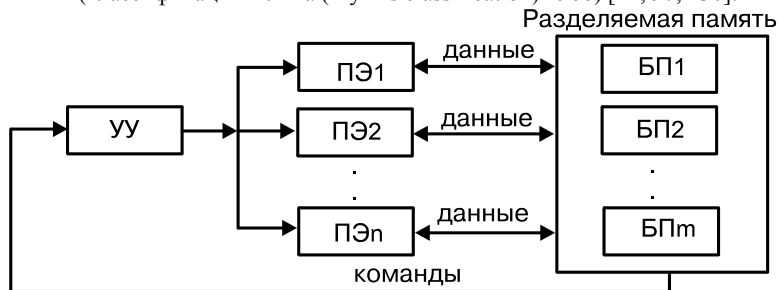


Рис. 1.6.а Структура ВС - SIMD

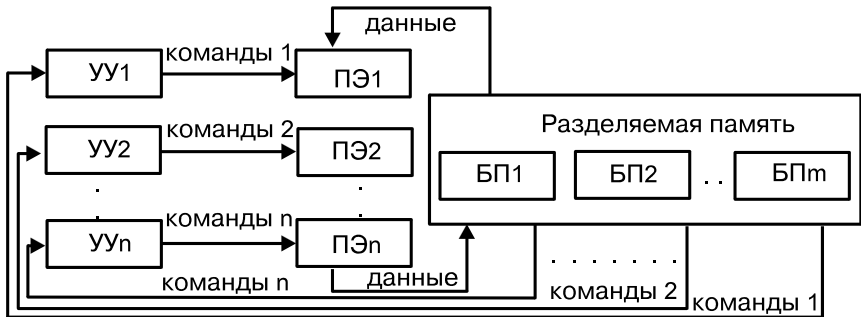


Рис. 1.6.б Структура BC - MISD

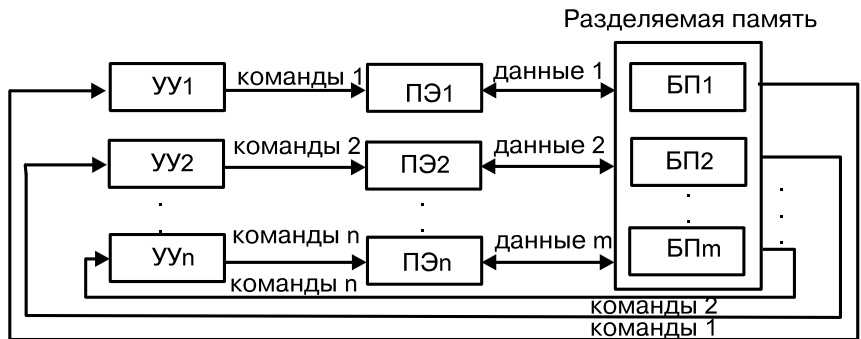


Рис. 1.6.в Структура BC - MIMD

В табл. 1.1 приведены некоторые наиболее известные коммерческие параллельные и последовательные BC.

Табл. 1.1

SISD с одним функциональным блоком	IBM 701, IBM 1620, IBM 7090, PDP VAX 11/780
SISD с множеством функциональных блоков (n)	IBM 360/91, IBM 370/168UP, CDC 6600, CDC Star-100, TI-ASC, FPS AP-120B, FPS AP-164, IBM 3838, Cray-1, CDC Cyber-205, Fujitsu VP-200, CDC-NASF, Fujitsu FACOM-230/75
SIMD	Illiac-IV, PEPE, BSP, STARAN, MPP, DAP, MasPar, MP

MIMD	IBM 370/168 MP, Univac 1100/80, Tandem/16, IBM 3081/3084, Cm*, Burroughs D-825, C.mmm, Cray-2, S-1, Cray-x MP, Denelcor HEP, IBM SP, Intel Paragon, CM5, Cray T3D, Meiko CS - 2, nCUBE Silicon Graphics Challenge, Sequent Symmetry.
------	--

Тзе-Йан Фенг в 1972 г. (Tse-Yun Feng) предложил использовать степень параллелизма для классификации различных архитектур ВС, а Хэндлер (Haendler) в 1977 г. предложил новую классификацию ВС, учитывающую степень параллелизма и степень конвейеризации вычислений.

В параллельных ВС исполнительная среда позволяет осуществлять параллельное выполнение нескольких команд, однако, при этом может потребоваться одновременный параллельный доступ к памяти. Следовательно, необходим более сложный механизм взаимодействий исполнительных вычислительных узлов с модулем или модулями памяти для выполнения операций над данными или получения инструкций. По этому признаку архитектуры параллельных ВС можно разделить на два класса:

- * ВС, допускающие одновременный и случайный доступ к общей (глобальной и локальной) памяти. Некоторые из них относятся к PRAM ВС (машины с параллельным случайным доступом). С точки зрения архитектуры – это компьютеры с разделяемой памятью.
- * ВС, в которых каждый процессор имеет прямой или случайный доступ только к своей локальной памяти и, кроме того, может косвенно — через сеть (interconnection network) обратиться к памяти других процессоров.

Если сеть статична, мы имеем дело с компьютером FITM (машины с постоянной топологией), в противном случае это FLTM (машины с динамической топологией или реконфигурируемые).

Концепция сети связи заимствована из PRAM модели, исходя из предположения о том, что доступ к памяти выполняется в один и тот же момент времени. В реальных машинах, базирующихся на концепции PRAM архитектуры, необходимы мощные средства связи с высокой пропускной способностью, что на практике не достижимо потому, что доступ к разделяемой памяти происходит через коммуникационную сеть и требует высоких аппаратных затрат. В настоящее время активно ведутся разработки высокопроизводительных коммуникационных систем передачи информации для параллельных вычислительных систем. За последние пятнадцать лет скорость передачи информации в распределенных системах обработки информации увеличилась на несколько порядков (рис. 1.7). Среди получивших наибольшее распространение сетевых технологий можно отметить следующие [89]:

- Ethernet — имя данное наиболее популярной технологии построения локальной сети, разработанной компанией Xerox PARC. Ethernet это шинная технология

передачи информации с распределенным управляемым доступом со скоростью передачи до 10 Мбит/с.

- FDDI — Fiber Distributed Data Interface. FDDI — это ВС с кольцевой архитектурой, использующая оптоволоконный канал связи между станциями и двойное кольцо для обеспечения надежности.
- HiPP — High-performance parallel interface. HiPP — это стандарт передачи данных на 800 Мбит/с по 32 или на 1.6 Гбит/с по 64 параллельным медным линиям. Большинство распространенных коммерческих ВС используют HiPP интерфейс.
- SONET — Synchronous Optical Network. SONET — это серия оптических сигналов на основе умножения базовой скорости 51.84 Мбит/с названной OC-1. OC-3 (155.52 Мбит/с) и OC-12 (622.08 Мбит/с) были разработаны для B-ISDN сетей. Кроме того, уже определен стандарт OC-192 (9.952 Гбит/с) для будущих сетей.

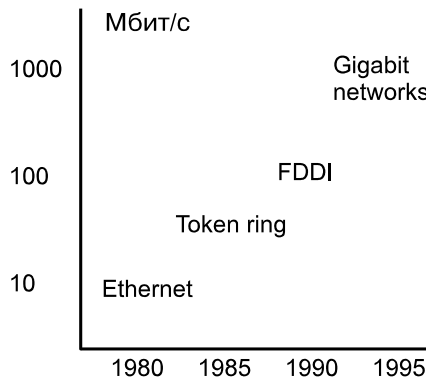


Рис. 1.7 Скорости передачи информации в сетях

- ATM — Asynchronous Transfer Mode. ATM является технологией для передачи, мультиплексирования и переключения, что обеспечивает высокую степень гибкости, требуемой в B-ISDN. ATM — это протокол связи, который использует пакеты с фиксированным 48 байтовым размером имеющим 5-ти байтовые заголовки.

Однако, для реализации PRAM архитектур требуется достаточно сложная технология организации вычислительного процесса для реализации концепции конкурирующего доступа, даже в том случае, когда для одновременной множественной операции чтения в разделяемой памяти используется секционирование памяти или ассоциативная память.

Обе эти модели (FITM, FLTM) предполагают, что обращение к памяти осуществляется в одно и то же время, независимо от способа организации памяти.

В любом случае, приведенные ниже различные модели параллельных машин описываются одними и теми же характеристиками абстрактной модели абстрактной алгебры.

Представим наиболее важные особенности этих классов параллельных машин.

1.3.1. Модель с разделяемой памятью (PRAM модель)

На рис.1.8 приведена структура PRAM модели. NRAM процессоров P_i [$0 \leq i \leq (N-1)$] могут иметь случайный и одновременный доступ к общей разделяемой памяти в одно и то же время.

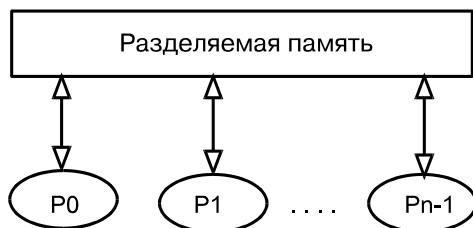


Рис. 1.8

Эта модель демонстрирует проблему доступа, когда несколько компьютеров одновременно обращаются к одной и той же ячейке памяти. Если одновременность выполнения операций чтения кажется концептуально корректной (даже если физически эта реализация не очевидна), то это не относится к совмещению операций записи. Поэтому оценка параллельного выполнения операций записи и/или чтения приводит к разделению BC PRAM архитектуры на несколько подклассов организации и управления памятью: SRSW, SRMW, MRSW и MRMW.

* SRSW модель организации памяти предусматривает, что только один процессор имеет право на операцию чтения или записи в одну ячейку памяти;

- * SRMW модель допускает один процессор для чтения и множество процессоров для записи;
- * MRSW модель допускает одновременное чтение множества данных, но только единственную операцию записи.
- * MRMW модель позволяет обоим операциям чтения и записи выполняться одновременно несколькими процессорами.

SRSW и MRMW наиболее популярны при исследовании параллельных алгоритмов и организации процессов в параллельных системах, а PRAM SRSW — для технической реализации. SRSW модели простые и доступ к памяти в них может быть реализован в виде RAM модели.

В случае машин, допускающих параллельную, но конкурентную операцию записи в память, понятие параллельности записываемых в область памяти данных не всегда точно определено. Несколько процессоров одновременно требуют выполнения операции записи множества данных в память. Здесь можно определить некоторые наиболее распространенные реализации этого "параллелизма".

- * детерминированный — это означает, что предпочтение на запись данных отдается процессору с наивысшим приоритетом;
- * комбинированный — это означает, что данные некоторых процессоров объединены путем сокращения операций (обычно ассоциативных и коммутативных, таких как сложение, умножение, max, or,...) и уменьшено количество обращений к памяти за счет хранения промежуточных результатов в процессоре (только результат операции передается в память);
- * недетерминированный — непредсказуемый, когда произвольное значение должно быть немедленно запомнено в памяти.

1.3.2. Вычислительные системы со статической (FITM) топологией

ВС с фиксированной топологией (FITM) предполагают наличие жесткой сети, которая связывает различные процессоры. Доступ процессоров к памяти других процессоров в таких машинах осуществляется через промежуточные процессоры с использованием сетевой связи, к собственной же памяти процессоры имеют прямой доступ.

Рис.1.9. в общих чертах представляет структуру архитектуры FITM ВС. Сеть связи соединяет n RAM процессоров P_i ($0 \leq i \leq (n-1)$), каждый из которых имеет свою собственную память со случайным доступом M_i .

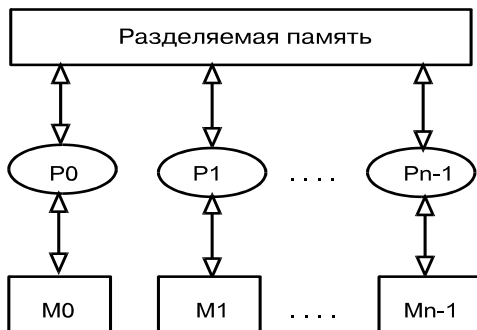


Рис. 1.9

Некоторые коммерческие компьютеры (СМ, Sphinx, MasPar's MP-1) поддерживают именно эту модель, которую значительно легче реализовать, чем общую PRAM модель.

В FITM модели обмен данными между вычислительными узлами играет важную роль и в значительной степени определяет характеристики ВС. В этом случае описание компьютерной модели (1) модифицируется в форму (2):

$$\text{компьютер} = \text{данные} + \text{операции} + \text{обмен данными} \quad (2)$$

Для соседних ячеек памяти статическая сетевая топология допускает непосредственный обмен. Для осуществления глобального обмена используют механизм, называемый трассировкой, допускающий обмен данными через связывающую сеть определенной топологии (матричная, гиперкуб, бинарное дерево, пирамидальная и т.д.).

В случае SIMD FITM (только коммерческих) возможна синхронизация общих связей (выполняемая параллельно с вычислениями, например Connection Machine). В случае MIMD FITM возможен как синхронный, так и асинхронный обмен данными.

Синхронизация связи последовательных процессов поддерживается на основе принципа Гоара [98] (Hoare's principle), который требует предварительной синхронизации процессоров перед обменом данными. Этот механизм часто использовался в транспьютерных системах.

Асинхронный обмен данными возможен, но требует существования канала связи. Это механизм нашел применение в пирамидальных компьютерах SPYINX [58] и в компьютерах MEDECINE.

Статические системы (FITM) часто используются в системах с распределенной памятью. Имеются следующие типы систем этого класса:

- Полносвязанная сеть (Completely-Connected Network), где каждый вычислительный узел имеет прямую связь с другим (рис. 1.10).

- Звезда (Star-Connected Network), где имеется один центральный процессор, который связан со всеми другими процессорами и через него осуществляется связь между ними (рис 1.11).
- Линейная и Кольцевая сети (Linear Array and Ring), где каждый вычислительный узел связан с двумя соседними узлами (рис 1.12.а, рис. 1.12.б).
- Петля (сеть) (Mesh Network). Эта сеть может быть одномерной (Кольцевая), двумерной — матричная (рис 1.13 а), ... и также с обратной связью — торроидальная (рис 1.13.б). Примерами двумерной системы являются DAP и Paragon XP/S, трехмерной системы (рис. 1.13.в). являются Cray T3D, Tera computer, и J-Machine.
- Дерево (Tree Network), где существует только одна связь между каждой парой вычислительных узлов. Примеры такой системы является DADO. Для увеличения способности передачи имеется, так называемое, жирное дерево (рис 1.14) в CM-5.



рис 1.12.а. Линейная

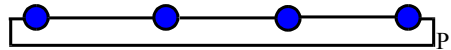


Рис 1.12.б. Кольцевая

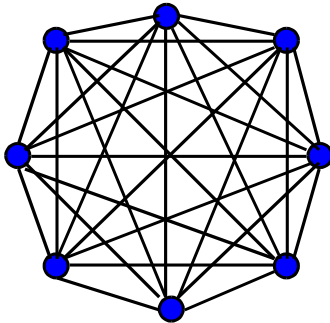


Рис 1.10. Полносвязная

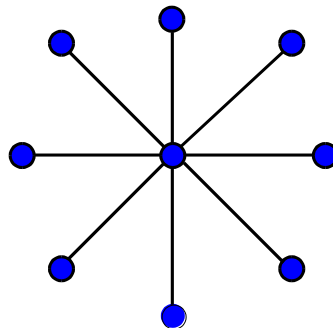


Рис 1.11 Звезда

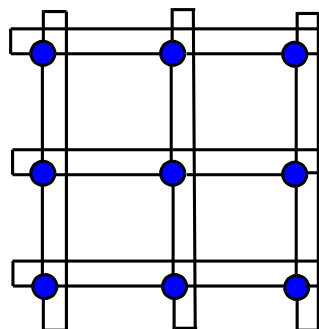
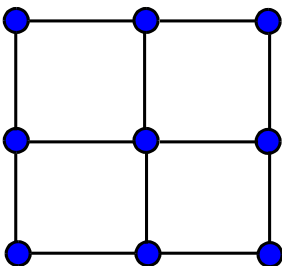


Рис 1.13.а. Матричные

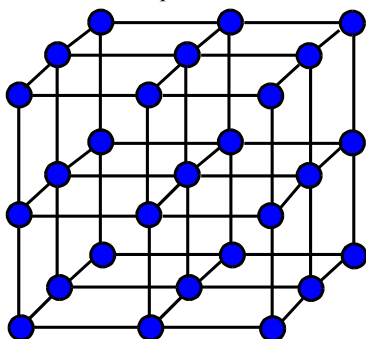


Рис 1.13.б. Торроидальные

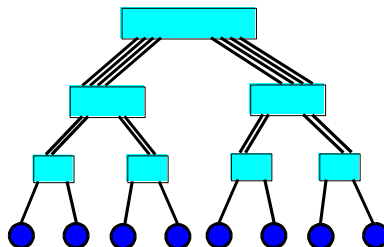


Рис. 1.13.в. Трехмерная сеть

- Гиперкуб (Hypercube Network). Это многомерная сеть процессоров с двумя процессорами в каждом измерении (рис. 1.15.а-г). D-мерный гиперкуб содержит $p=2^D$ процессоров. Примерами такой системы являются nCUBE 2, Cosmic Cube, и iPSC.
- k-d Сеть (k-ary d-cube Networks). Эти топологии определяют пределы класса топологий, называемых k-ary d-cube. Где D размерность сети, а K — основание системы счисления сети, которое определяется как количество процессоров в каждом измерении. Число процессоров в сети равна K^D . d-мерный гиперкуб также называемый бинарным d-кубом — это d-мерная петля с 2-мя процессоров в каждом измерении.

Рис 1.14. Жирное дерево

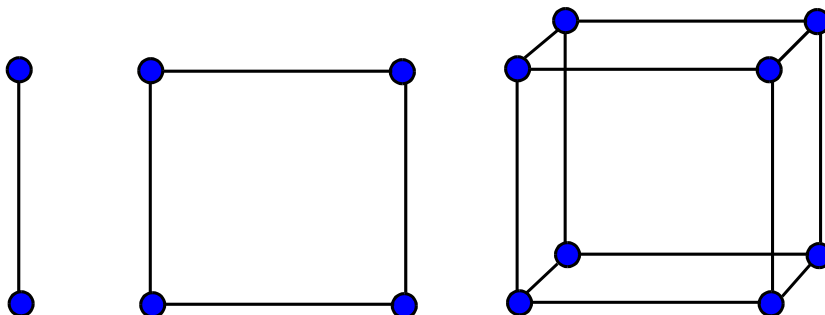


Рис 1.15.а. 1-D Рис 1.15.б. 2-D гиперкуб Рис 1.15.в. 3-D гиперкуб

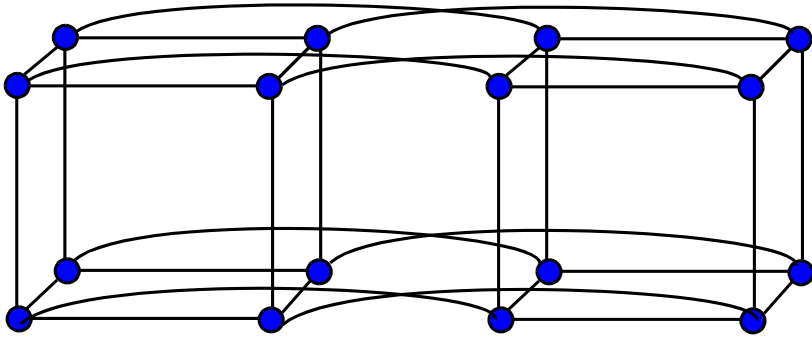


Рис 1.15.г. 4-D гиперкуб

Табл. 1.2

Сравнение характеристик статических сетей с p процессорами

Вычислительные системы	Диаметры	Бисекторная ширина	Связность по дугам	Число каналов
Полносвязная сеть	1	$p^2/4$	$p-1$	$p(p-1)/2$
Звезда	2	1	1	$p-1$
Линейная	$p-1$	1	1	$p-1$
Кольцевая	$\lceil p/2 \rceil$	2	2	p
Матричная	$2\lceil \sqrt{p}/2 \rceil$	$2\sqrt{p}$	4	$2p$
Древовидная	$2\log((p+1)/2)$	1	1	$p-1$
Гиперкуб	$\log p$	$p/2$	$\log p$	$(p \log p)/2$
k -d сеть	$d\lceil k/2 \rceil$	$2k^{d-1}$	$2d$	dp

- * **Диаметр** сети — это максимальное расстояние между любыми двумя процессорами в сети. Расстояние между двумя процессорами определяется как ближайший путь (в количестве соединений) между ними. Поскольку большее расстояние определяет время связи, сети с меньшим диаметром лучше.
- * **Связность** сети — это мера сложности пути между любыми двумя процессорами. Желательно, чтобы сеть была с высокой связностью, т.к. это снижает число соединений для коммутации ресурсов. Одно из измерений связности — это число дуг, которое должно быть удалено из сети, чтобы превратить ее в две взаимонесвязанные сети. Это называется **связность по дугам**.
- * **Бисекторная ширина** и **бисекторная полоса пропускания**. Бисекторная ширина сети определяется как минимальное число связей, которое должно быть

удалено, чтобы разбить сеть на две равные половины. Число бит, которые могут передаваться одновременно через соединение двух процессоров, называется **шириной канала**. Ширина канала равна числу физических шин в каждом канале связи. Пиковая частота на которой одиночный физический носитель может выдавать (передавать без ошибки) биты называется **частотой канала**. Пиковая частота, на которой данные могут передаваться между концами канала связи, называется **полосой пропускания канала**. Полоса пропускания канала зависит от частоты канала и шириной канала

- * **Себестоимость (число каналов)**. Много критериев может быть использовано для оценки себестоимости сети. Один из методов определения себестоимости сети — число каналов связи или число физических носителей в сети.

1.3.3. Вычислительные системы с динамической (FLTM) топологией

Рассмотрим параллельный FLTM компьютер PRAM модели с **p** процессорами и общей памятью, состоящей из **m** слов. Процессоры соединяются с памятью через множество переключаемых элементов. При этом обеспечивается доступность слова памяти каждому процессору. Любой из **p** процессоров может обратиться к любому слову памяти, обеспечивая при этом единичность своего доступа. Для обеспечения этого условия общее число переключаемых элементов, в общем случае, должно быть пропорционально произведению **m** \times **p**. Отсюда видно, что даже при небольших размерах памяти построение переключаемой сети такой сложности является очень дорогостоящим. Один из способов устранения данного недостатка состоит в разбиении всей памяти на отдельные участки, называемые банками. При этом процессоры системы будут переключаться между банками. Однако и этот способ не лишен недостатков. При обращении процессора к одному из банков все остальные процессоры обратиться туда уже не могут. В параллельных компьютерах применяются следующие способы организации взаимодействия процессоров и банков памяти:

- * Перекрестные переключаемые сети. (Crossbar switching Networks)
- * Шинно-ориентированные сети. (Bus-Based Networks)
- * Многоступенчатые взаимосвязанные сети (Multistage Interconnection Networks)

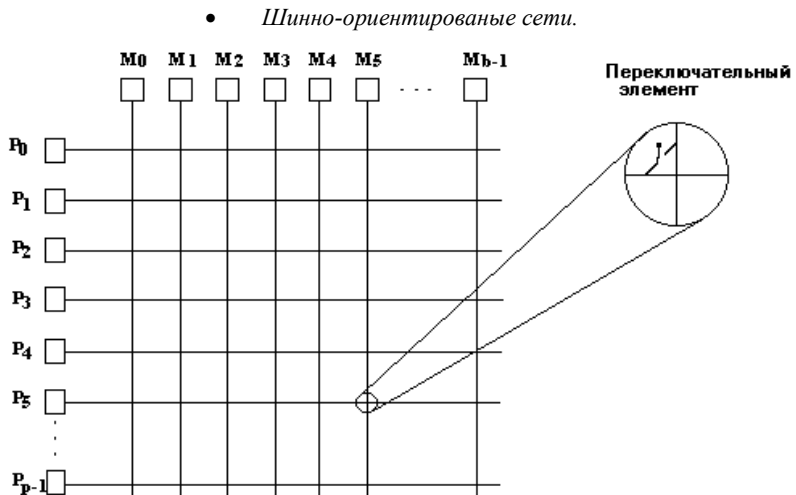
- *Перекрестные переключаемые сети*

Наиболее простой способ соединения **p** процессоров и **b** банков памяти получается при использовании перекрестных переключателей. Перекрестный переключатель (рис. 1.16) использует решетку переключаемых элементов.

Перекрестные переключаемые сети являются неблокирующими сетями в том смысле, что соединение одного процессора с банком памяти не препятствует другим процессорам соединяться с любыми другими банками. Обычно число банков **b** берется большим или равным числу процессоров, так что каждый процессор в худшем случае имеет один банк памяти для доступа. Общее число

переключательных элементов, требуемых для построения такой сети, равно $p \times b$. Это приемлемо, если предположить, что число банков по крайней мере равно p . Если в системе $b < p$, то тогда в любой момент времени могут появиться процессоры не имеющие доступа к памяти. Отсюда следует, что с увеличением числа процессоров p сложность сети возрастает как p^2 . Поэтому переключательные сети становятся труднореализуемыми при большом числе процессоров, при этом значительную роль играет также и ценовой фактор. Перекрестные переключательные сети в настоящее время используются в таких системах как Cray-YMP и Fujitsu VPP 500. В VPP 500 используется перекрестная сеть размером 224×224 .

Рис.1.16. Перекрестная переключательная сеть



В шинно-ориентированных сетях процессоры соединяются с общей памятью посредством единого канала данных, называемого шиной. Такая система очень проста для реализации. Рис. 1.17(а) иллюстрирует обычную шинную архитектуру. При обращении процессора к общей памяти на определенных линиях шины выставляются соответствующие сигналы.

Главным недостатком данного способа является то, что при увеличении числа процессоров, связанных с общей шиной, весьма значительно увеличивается время ожидания каждым процессором доступа к памяти. Данный способ пригоден для небольшого числа процессоров.

Один из способов решения данной проблемы заключается в обеспечении каждого процессора локальной кэш-памятью, как показано на рис.1.17(б).

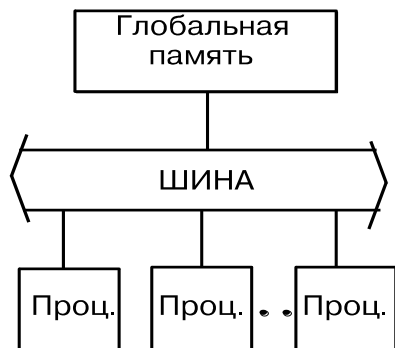


Рис.1.17.а

Шинно-ориентированные сети

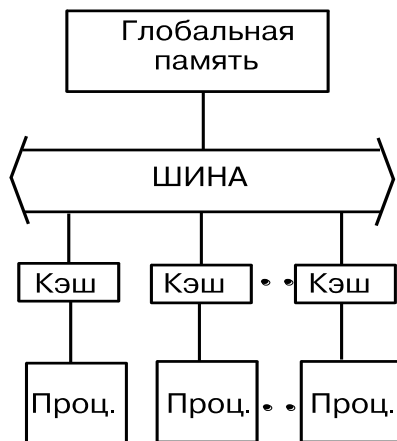


Рис. 1.17.б

Кэш шинно-ориентированные сети

Использование памяти в шинно-ориентированных сетях с FLTM топологией отличается от FITM BC. При обычных вычислениях, когда ссылка делается на какой либо фрагмент памяти, следующая ссылка вероятней всего будет сделана на фрагмент памяти, следующий за этим фрагментом. Этот вывод сделан на основе свойств "временной" и "пространственной" локальности теории рабочих множеств. Следовательно, весь этот фрагмент может быть помещен в локальную кэш-память процессорного элемента BC и всю дальнейшую обработку данных он может вести не обращаясь к общей памяти. При этом существенно сокращается время работы процессора с общей памятью. В случае кэш-промаха (т.е. когда в локальной памяти не оказалось требуемого слова) процессору необходимо еще раз обратиться к общей памяти. Данный способ применяется в машинах Symmetrie, Multimax и других.

- *Многоступенчатые взаимосвязанные сети*

Перекрестные переключательные сети выигрышны с точки зрения производительности, но не всегда приемлемы из-за высокой цены. Наоборот, шинно-ориентированные сети приемлемы по стоимости, но имеют ограничения по производительности. В качестве среднего класса сетей используются многоступенчатые взаимосвязанные сети. Они более производительные чем

шинные и дешевле перекрестных. Рис.1.18(а) [97] иллюстрирует зависимость стоимости перекрестных, многоступенчатых, шинно-ориентированных сетей в зависимости от числа процессоров. На рис. 1.18(б) [97] показана зависимость производительности данных сетей от числа процессоров.

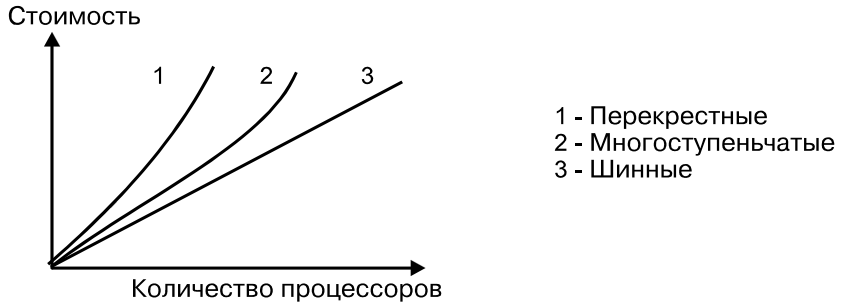


Рис.1.18(а). Зависимость стоимости реализации от числа процессоров

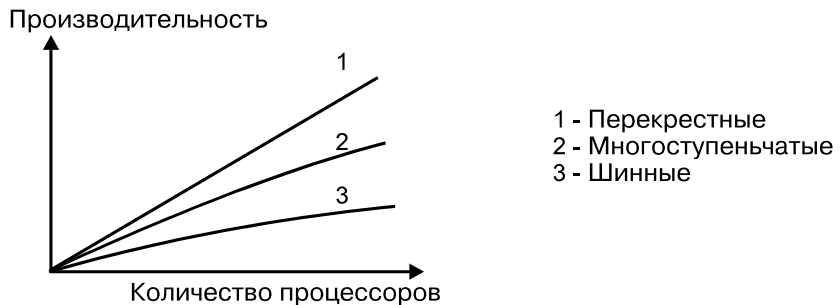


Рис.1.18(б). Зависимость производительности от числа процессоров

Обычно многоступенчатая сеть включает **p** процессоров и **b** банков памяти, как показано на рис.1.19.

Многоступенчатая взаимосвязанная сеть обычно использует для связи ступеней *delta* сеть. В сети присутствует $\log p$ ступеней, где **p** число процессоров и банков памяти. *Delta* сеть включает взаимосвязанную модель, содержащую **p** входов и **p** выходов. Между входом *j* и выходом *i* есть связь при выполнении условия:

$$J = \begin{cases} 2^i, & \text{т } \text{д } 0 \leq i \leq p/2 - 1 \\ 2^i + 1 - p, & \text{т } \text{д } p/2 \leq i \leq p - 1 \end{cases} \quad (2.1)$$

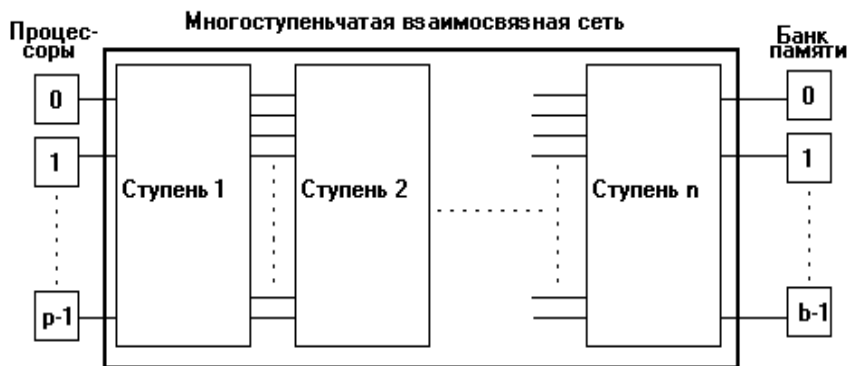


Рис.1.19. Многоступенчатая взаимосвязанная сеть

Эта модель связи называется "полная перестановка" (Perfect shuffle). Рис.1.20 иллюстрирует полную модель для 8 входов и выходов. В каждой ступени *delta* сети полная перестановочная модель связи включает $p/2$ переключаемых элементов. Каждый переключатель может находиться в одном из двух состояний. В первом состоянии входы связываются с выходами напрямую, как показано на рис.1.21(а). Это называется прямой связью. В другом состоянии входы соединяются с выходами перекрестным способом, как показано на рис.1.21(б). Это называется перекрестной связью. Возможны еще два варианта связей рис.1.21(в,г).

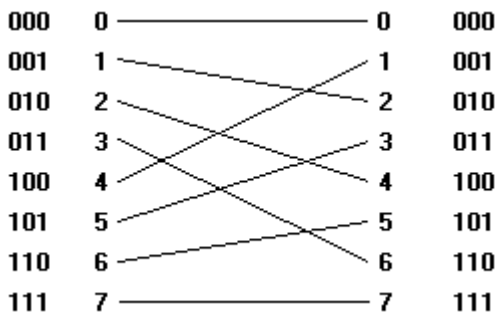


Рис. 1.20

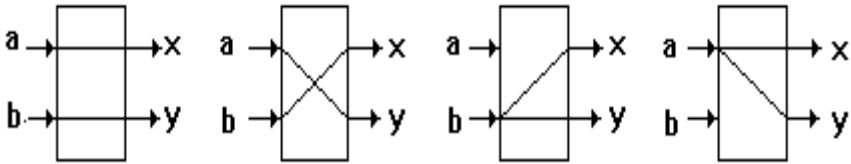


Рис.1.21(а)

1. 21(б)

1. 21(в)

1.21(г)

Вся *delta* сеть имеет $p/2 \times \log(p)$ переключающих элементов и сложность такой сети растет как $p \times \log(p)$. Заметим, что эта сложность ниже чем p^2 сложности перекрестной сети. Рис. 1.22 показывает *delta* сеть для 8-ми процессоров. Процессоры расположены на входе сети, банки памяти - на выходе. Передача сообщений в *delta* сети выполняется по простой схеме. пусть s и t двоичное представление источника и приемника сообщения. Сообщение пересекает звено первого переключающего элемента. Если старшие биты в s и t одинаковы, тогда сообщение передается по прямой связи переключателя. Если эти биты различны, то тогда сообщение передается по перекрестной связи переключателя. Это правило повторяется в следующих ступенях, где для анализа используются следующие младшие разряды. При пересечении $\log(p)$ ступеней используются все $\log(p)$ биты в двоичном представлении s и t .

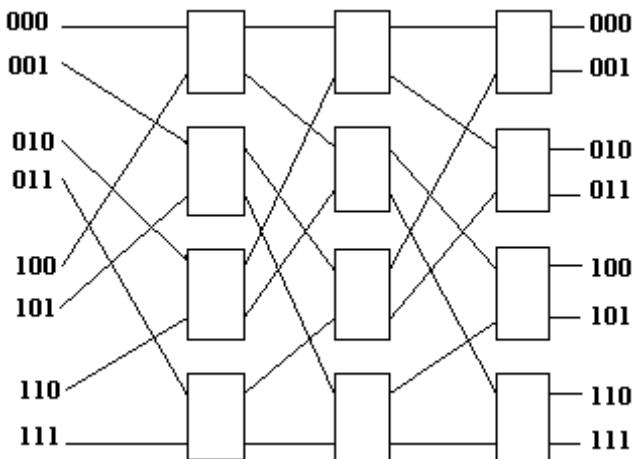


Рис.1.22 Delta сеть для 8-ми процессоров

На рис.1.23 показана передача сообщения в 8-ми процессорной *delta* сети из процессора 2(010) в банк 7(111) и из процессора 6(110) в банк 4(100). Этот рисунок

иллюстрирует важное свойство *delta* сети. Когда процессор 2(010) связывается с банком 7(111), он блокирует путь из процессора 6(110) в банк 4(100). Связывающее звено АВ входит в оба пути. Отсюда видно, что в *delta* сети обращение одного процессора к банку памяти может запретить обращение другого процессора к другому банку памяти. Сети с этим свойством относятся к *блокирующим сетям* (blocking networks).. На основе *delta* сети строятся параллельные компьютеры BBN Butterfly, IBM RP-3, NYU Ultracomputer.

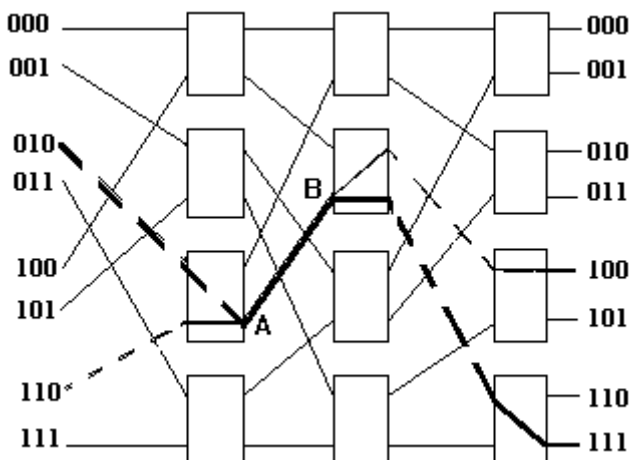


Рис. 1.23. Блокирование пути одним сообщением

1.4. ЗАКЛЮЧЕНИЕ

На рис. 1.24. приведена классификация различных существующих моделей компьютеров, а в табл. 3 [106] некоторые наиболее известные суперкомпьютеры и минисуперкомпьютеры.

RAM и PRAM модели соответствуют уравнению (1).

RAM модель, которая поддерживается компьютерами с жесткой последовательностью операций (последовательные), может быть рассмотрена как SRSW машина.

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

машины с прямым доступом к памяти

ВС со случайным доступом к

		локальной памяти и косвенным доступом к локальной памяти других машин	
RAM	PRAM	FITM	FLTM
последовательные машины	машины с разделяемой памятью	машины с жесткой топологией	машины с динамической топологией
		с распределенной памятью	
		SRSW	гиперкуб
		SRMW	полносвязная
		CCC	ОМЕГА
		MRSW	древовидная
		MRMW	Cray-YMP
		пирамидальная	VPP 500
		кольцевая	Symmetre
		звезда	BBN Butterfly
		* * * * *	IBM RP-3
компьютер=данные+операции		компьютер=данные + операции + обмен данными	

Рис. 1.23. Компьютерные модели

PRAM модель реализована в ВС с разделяемой памятью. В этих машинах допускается одновременный доступ к общей разделяемой памяти непосредственно через сеть связи. Но, несмотря на всю привлекательность математической организации PRAM модели, реально не было создано ни одного процессора, поддерживающего эту модель в полном объеме с соответствующими функциональными требованиями.

Модель FITM используется в компьютерах с распределенной памятью. В таких компьютерах реализуется случайный доступ к собственной локальной памяти каждого процессора и косвенный доступ к памяти других процессоров с помощью транзитной пересылки через связи с соседними процессорами или с помощью общей сети связи.

Табл. 1.3

Компьютер	Фирма производитель	Количество процессоров	Мах. производ.	Операц. система
-----------	------------------------	---------------------------	-------------------	--------------------

			(Мфлорс)	
CRAY-1	Cray Reseach	1	160	COS
CRAY X-MP	Cray Reseach	1,2,4	210/пп	COS, Unix
CRAY-2	Cray Reseach	4	2000	Unix
CRAY Y-MP	Cray Reseach	8	2666	COS, Unix
CRAY Y-MPC-90	Cray Reseach	16	16000	Unix
CYBER 205	CDC	1	400	VSOS
ETA ¹⁰	ETA Systems	2,4,6,8	800/пп	VSOS, Unix
FACOM VP-400	Fujitsu	1	1142	MVS
S-810/20	Hitachi	1	630	MVS
SX-2	NEC	1	1333	ATSS-AF
SX-3	NEC	1,2,4	5500/пп	SXOS, Unix
FX/8	Alliant Comp.	8	94	Unix
C-1	Convex	1	60	Unix
SCS-40	ScientificComp	1	44	COS
FPS Series T	Floating Point	16-16384	16/пп	Unix
VAX 9000VP	DEC	4	500	VMS, Unix
DECmpp 12000	DEC	1024-16384	1200	Unix
Ametec 2010	Symult Systems	4-512	215	React. Kernel
GP 1000	BBC ACI	2-256	*	Chrysalis
CM-2	Thinking Mach	65536	2400	Unix
CM-5	Thinking Mach	1024-16384	130000	Unix
ELXSI 6400	ELXSI	1-12	14	Unix
Multimax	Encore	2-20	20	Unix
MPP	Goodyear	16384	470	*
Sequent	Sequent omp.	2-30	*	Unix
iPSC/2	Intel	32-128	1000	Unix
iPSC/860	Intel	8-128	7680	NX/2
Paragon	Intel	64-4096	300000	Unix
NCube/10	NCube	4-1024	500	Vertex
NCube/2	NCube	32-8192	27000	Vertex
SuperNode 1000	Parsys	16-1024	1600	IDRIS
VOLVOX	Archipel	8-48	3000	Unix
GC	Parsytec	64-16384	400000	Unix

* данные не известны

Некоторые FITM машины были реализованы. Наиболее популярна топология связи процессорных элементов в виде сетки процессоров (Iliac IV/NASA, Clip4/Univ. College of London, MPP/GoodYear, DAP/ICL, MP1/MasPar, SYMPATI,...). CM-1 компании Thinking Mashine Corporation, iPCS — это пример машины с гиперкубической топологией размерностью 12. BC CM вначале представлялась SIMD моделью, а позже как MIMD. Также достаточно хорошо была изучена пирамидальная структура; в компьютере Sphinx/Univ. в ParisXI–France реализована бинарная структура с мульти–SIMD управлением, остальные структуры – кватернарные (четверичные) только с SIMD управлением, например PAPIA/Univ. of Pavia–Italy и GAM/Georges Mason Univ.–USA. Колумбийским Университетом (Columbia University/ New–York) были исследованы три вида топологии связей в параллельных не фон Нэймановских машинах, предназначенных для обработки баз данных.

Таким образом, изучение архитектуры, организации параллельных вычислительных систем, а так же тенденций развития показывает, что:

1. Параллелизм в настоящее время является вездесущим, он присутствует и в проектировании BC и в организации вычислительных процессов.
2. Достижение современной технологии в коммуникации, таких как FDDI, HiPP или SONET позволяет строить мощные вычислительные системы распределенных ресурсов, где можно обрабатывать разнотипные задачи.
3. Задачи планирования играют особую роль в повышении эффективности вычислений систем любого типа.
4. Тенденции развития BC требует нового осмысления понятия Заданий и Ресурсов (объекты задачи планирования), а так же новую схему для их распределения.
5. Масштаб современных BC, особенности их конфигурации и управления, усложнил процесс планирования. Поэтому, для получения эффективные решения нужно рассматривать задачи планирования на уровне ее составляющих (подзадач).