

А. М. Ларионов, С.А. Майоров,

Г. И. Новиков

ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ, СИСТЕМЫ И СЕТИ

Допущено Министерством высшего и среднего
специального образования СССР
в качестве учебника для студентов вузов,
обучающихся по специальности
«Электронные вычислительные машины»

Ленинград
ЭНЕРГОАТОМИЗДАТ
Ленинградское отделение
1987

Оглавление

ПРЕДИСЛОВИЕ	3
ГЛАВА ПЕРВАЯ	5
СИСТЕМЫ ОБРАБОТКИ ДАННЫХ	5
1.1. СПОСОБЫ ПОСТРОЕНИЯ И КЛАССИФИКАЦИЯ	5
1.2. СОСТАВ И ФУНКЦИОНИРОВАНИЕ	12
1.3. ХАРАКТЕРИСТИКИ И ПАРАМЕТРЫ	17
1.4. РЕЖИМЫ ОБРАБОТКИ ДАННЫХ	24
ГЛАВА ВТОРАЯ	29
ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ	29
2.1. ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ИНФОРМАЦИИ	29
2.2. МНОГОМАШИННЫЕ КОМПЛЕКСЫ	37
2.3. МНОГОПРОЦЕССОРНЫЕ КОМПЛЕКСЫ	40
2.4. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ	46
2.5. КОМПЛЕКСЫ НА ОСНОВЕ МИКРО-ЭВМ И МИКРОПРОЦЕССОРОВ	51
2.6. РЕАЛИЗАЦИЯ КОМПЛЕКСОВ	54
2.7. СРАВНЕНИЕ МНОГОМАШИННЫХ И МНОГОПРОЦЕССОРНЫХ КОМПЛЕКСОВ	62
ГЛАВА ТРЕТЬЯ	66
ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ	66
3.1. СИСТЕМЫ С КОНВЕЙЕРНОЙ ОБРАБОТКОЙ ИНФОРМАЦИИ	66
3.2. МАТРИЧНЫЕ СИСТЕМЫ	70
3.3. АССОЦИАТИВНЫЕ СИСТЕМЫ I	75
3.4. ОДНОРОДНЫЕ СИСТЕМЫ И СРЕДЫ	78
3.5. ФУНКЦИОНАЛЬНО РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ	80
3.6. СИСТЕМЫ С ПЕРЕСТРАИВАЕМОЙ СТРУКТУРОЙ	83
ГЛАВА ЧЕТВЕРТАЯ	88
СИСТЕМЫ ТЕЛЕОБРАБОТКИ	88
4.1. ПРИНЦИПЫ ПОСТРОЕНИЯ	88
4.2. КАНАЛЫ СВЯЗИ	90
4.3. СОПРЯЖЕНИЕ ЭВМ С КАНАЛАМИ СВЯЗИ	96
4.4. АБОНЕНТСКИЕ ПУНКТЫ	97
4.5. ПРОГРАММНЫЕ СРЕДСТВА	99
ГЛАВА ПЯТАЯ	101
ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ	101
5.1. СТРУКТУРА И ХАРАКТЕРИСТИКИ	101
6.2. МОНОКАНАЛЫ	106
6.3. АДАПТЕРЫ	117
6.4. РАСШИРЕНИЕ И КОМПЛЕКСИРОВАНИЕ	121
6.5. РЕАЛИЗАЦИЯ	123
ГЛАВА СЕДЬМАЯ	129
ОСНОВЫ ТЕОРИИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ	129
7.1. ПРЕДМЕТ И ЗАДАЧИ	129
7.2. МОДЕЛИ И МЕТОДЫ	133
7.3. ПРИНЦИПЫ АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ	148
7.4. МЕТОДЫ И СРЕДСТВА ИЗМЕРЕНИЙ И ОЦЕНКИ ФУНКЦИОНИРОВАНИЯ	153
7.5. МОДЕЛИ РАБОЧЕЙ И СИСТЕМНОЙ НАГРУЗКИ	160
ГЛАВА ВОСЬМАЯ	168
ПРОЕКТИРОВАНИЕ И ЭКСПЛУАТАЦИЯ	168
8.1. ОРГАНИЗАЦИЯ ПРОЕКТИРОВАНИЯ	168
8.2. СИСТЕМОТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ	169
8.3. ЭКСПЛУАТАЦИЯ	174
СПИСОК ЛИТЕРАТУРЫ	178

ПРЕДИСЛОВИЕ

В «Основных направлениях экономического и социального развития СССР на 1986–1990 годы и на период до 2000 года» вычислительная техника рассматривается как один из важнейших факторов интенсификации производства к повышению производительности труда. В двенадцатой пятилетке производство средств вычислительной техники возрастает в 2–2,3 раза, что значительно расширит масштабы применения ЭВМ и систем на их основе в научных исследованиях, инженерной деятельности, управлении производством, технологическим оборудованием и транспортом. В реализации заданий Энергетической программы СССР существенная роль отводится автоматизированным системам разведки полезных ископаемых, управления топливно-энергетическими комплексами и энергетическим оборудованием.

Эффект, достигаемый за счет применения вычислительной техники, возрастает при увеличении масштабов обработки данных, т. е. концентрации по возможности больших объемов данных и процессов их обработки в рамках одной технической системы. Крупномасштабные системы обработки данных можно создавать, повышая мощность ЭВМ или объединяя многие ЭВМ в вычислительные комплексы и сети. Комплексование средств вычислительной техники позволяет создавать широкую номенклатуру высокопроизводительных отказоустойчивых систем обработки данных и рассматривается как одно из перспективных направлений развития вычислительной техники.

Методы проектирования и эксплуатации вычислительных комплексов, систем и сетей разрабатываются в рамках теории вычислительных систем. При этом считается, что термин «вычислительная система» охватывает и вычислительные, и управляющие, и информационно-измерительные системы, построенные на основе ЭВМ, многомашинных и многопроцессорных комплексов и вычислительных сетей. Характерная черта теории вычислительных систем – системотехнический подход к исследованию. Основной принцип системотехники – представление системы как совокупности всех ее частей в тесной связи с окружающей средой и взаимообусловленности всех свойств системы, таких, как производительность, надежность и стоимость. Поэтому в теории вычислительных систем технические средства и программные (операционная система и прикладное программное обеспечение) рассматриваются во взаимосвязи и свойства системы оцениваются как совокупность свойств технических и программных средств.

Естественно, что при системотехническом подходе к сложным объектам, какими являются вычислительные системы, комплексы и сети, необходимо ограничивать глубину исследования их организации и процессов функционирования и в качестве элементов систем рассматривать процессоры, каналы ввода – вывода, запоминающие и периферийные устройства, а во многих случаях – ЭВМ, многопроцессорные и многомашинные комплексы. Этот уровень представления вычислительных систем, комплексов и сетей – наиболее характерная черта системотехнического подхода в вычислительной технике.

Теория вычислительных систем (комплексов и сетей) состоит из двух основных разделов: архитектуры систем и метрической теории систем. Первый раздел – связан с изучением архитектуры вычислительных систем – общей логической организации систем, определяющей процессы обработки данных и включающей в себя методы представления данных, состав, назначение и принципы взаимодействия технических средств и программного обеспечения. Исследования архитектуры вычислительных систем направлены на выявление способов построения систем с различными свойствами, рациональных вариантов распределения функций между техническими и программными средствами, состава системы и конфигурации связей между ними, способов организации вычислительных процессов. Метрическая теория вычислительных систем направлена на количественную оценку показателей, характеризующих организацию и функционирование систем. В рамках метрической теории исследуется влияние организации и режима

функционирования на производительность, надежность, стоимость и другие характеристики систем, а также решаются задачи выбора оптимальных технических характеристик устройств, входящих в состав проектируемых систем.

В данном учебнике основное внимание уделяется архитектуре вычислительных комплексов, систем и сетей, поскольку изучение логики их построения и функционирования – важнейшая составляющая подготовки инженеров – системотехников по ЭВМ. Метрическая теория вычислительных систем излагается применительно к задачам проектирования вычислительных комплексов, систем и сетей и представляется в виде конечных результатов. Желаящие детально изучить аппарат метрической теории вычислительных систем могут обратиться к работам [4, 5, 16–19].

Главы 1 и 4 написаны С. А. Майоровым, главы 2 и 3 (кроме § 3.5, 3.6) – А. М. Ларионовым, главы 5–8 и § 3.5, 3.6 – Г. И. Новиковым.

Отзывы о книге, замечания и пожелания просьба направлять по адресу: 191065, Ленинград, Марсово поле, 1, Ленинградское отделение Энергоатомиздата.

СИСТЕМЫ ОБРАБОТКИ ДАННЫХ

1.1. СПОСОБЫ ПОСТРОЕНИЯ И КЛАССИФИКАЦИЯ

Система обработки данных (СОД) – совокупность технических средств и программного обеспечения, предназначенная для информационного обслуживания пользователей и технических объектов. В состав *технических средств* входит оборудование для ввода, хранения, преобразования и вывода данных, в том числе ЭВМ, устройства сопряжения ЭВМ с объектами, аппаратура передачи данных, и линии связи. *Программное обеспечение* (программные средства) – совокупность программ, реализующих возложенные на систему функции. Функции СОД состоят в выполнении требуемых актов обработки данных: ввода, хранения, преобразования и вывода. Примерами СОД являются вычислительные системы для решения научных, инженерно-технических, планово-экономических и учетно-статистических задач, автоматизированные системы управления предприятиями и отраслями народного хозяйства, системы автоматизированного и автоматического управления технологическим оборудованием и техническими объектами, информационно-измерительные системы и др.

Основа СОД – это технические средства, так как их производительностью и надежностью в наибольшей степени определяется эффективность СОД.

Одномашинные СОД. Исторически первыми и до сих пор широко распространенными являются одномашинные СОД, построенные на базе единственной ЭВМ с традиционной однопроцессорной структурой. К настоящему времени накоплен значительный опыт проектирования к эксплуатации таких СОД, и поэтому создание, их, включая разработку программного обеспечения, не вызывает принципиальных трудностей. Однако производительность и надежность существующего парка ЭВМ оказывается удовлетворительной лишь для ограниченного применения, когда требуется относительно невысокая (до нескольких миллионов операций в секунду) производительность и допускается простой системы в течение нескольких часов из-за отказов оборудования. Повышение производительности и надежности ЭВМ обеспечивается в основном за счет совершенствования элементно-технологической базы. Достигнутое к настоящему времени быстроедействие электронных схем приближается к физическому пределу, и производительность ЭВМ на уровне десяти миллионов операций в секунду можно рассматривать как максимальную возможную. При любом уровне технологии нельзя обеспечить абсолютную надежность элементной базы, и поэтому нельзя для одномашинных СОД исключить возможность потери работоспособности. Таким образом, одномашинные

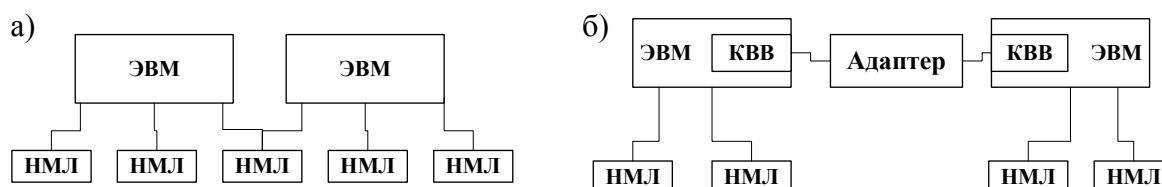


Рис. 1.1. Многомашинный вычислительный комплекс с косвенной (а) и прямой (б) связью между ЭВМ

Вычислительные комплексы. Начиная с 60-х годов для повышения надежности и производительности СОД, несколько ЭВМ связывались между собой, образуя *многомашинный вычислительный комплекс*.

В ранних многомашинных комплексах связь между ЭВМ обеспечивалась через общие внешние запоминающие устройства – накопители на магнитных дисках (НМД) или магнитных лентах (НМЛ) (рис 1,1, *а*), т.е. за счет доступа к общим наборам данных. Такая связь называется *косвенной* и оказывается эффективной только в том случае, когда ЭВМ взаимодействуют достаточно редко, например, при отказе одной из ЭВМ или в моменты начала и окончания обработки данных. Более оперативное взаимодействие ЭВМ достигается за счет *прямой связи* через адаптер, обеспечивающий обмен данными между каналами ввода – вывода ЧКВВ) двух ЭВМ (рис. 1.1, *б*) и передачу сигналов прерывания. За счет этого создаются хорошие условия для координации процессов обработки данных и повышается оперативность обмена данными, что позволяет вести параллельно процессы обработки и существенно увеличивать производительность СОД. В настоящее время многомашинные вычислительные комплексы широко используются для повышения надежности и производительности СОД.

В многомашинных вычислительных комплексах взаимодействие процессов обработки данных обеспечивается только за счет обмена сигналами прерывания и передачи данных через адаптеры канал – канал или общие внешние запоминающие устройства. Лучшие условия для взаимодействия процессов – когда все процессоры имеют доступ ко всему объему данных, хранимых в оперативных запоминающих устройствах (ОЗУ), и могут взаимодействовать со всеми периферийными устройствами комплекса. Вычислительный комплекс, содержащий несколько процессоров с общей оперативной памятью и периферийными устройствами, называется *многопроцессорным*. Принцип построения таких комплексов иллюстрируется рис. 1.2. Процессоры, модули оперативной памяти (МП) и каналы ввода–вывода, к которым подключены периферийные устройства (ПУ), объединяются в единый комплекс с помощью средств коммутации, обеспечивающих доступ каждого процессора к любому модулю оперативной памяти и каналу ввода–вывода, а также возможность передачи данных между последними. В многопроцессорном комплексе отказы отдельных устройств влияют на работоспособность СОД в меньшей степени, чем в многомашинном, т.е. многопроцессорные комплексы обладают большей устойчивостью к отказам. Каждый процессор имеет непосредственный доступ ко всем данным, хранимым в общей оперативной памяти, и к периферийным устройствам, что позволяет параллельно обрабатывать не только независимые задачи, но и блоки одной задачи.

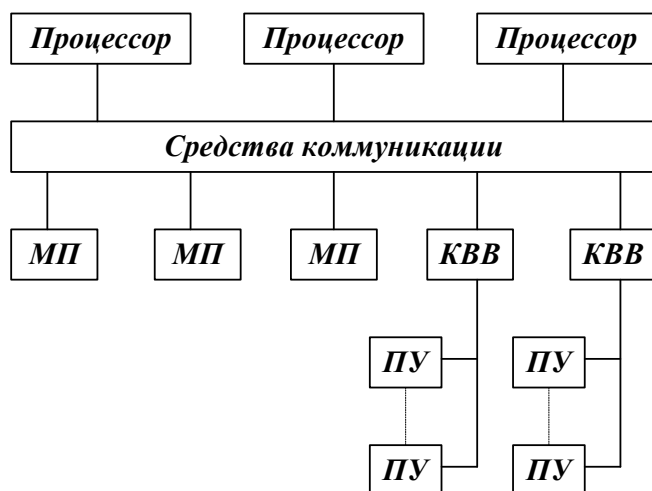


Рис. 1.2. Многопроцессорный вычислительный комплекс

Многомашинные и многопроцессорные вычислительные комплексы рассматриваются как базовые средства для создания СОД различного назначения. Поэтому в состав вычислительного комплекса принято включать только технические средства и общесистемное (базовое), но не прикладное программное обеспечение, связанное с конкретной областью применения комплекса. Таким образом, *вычислительный комплекс* – совокупность технических средств, включающих в себя несколько ЭВМ или процессоров, и общесистемного (базового) программного обеспечения.

Вычислительные системы. СОД, настроенная на решение задач, конкретной области применения, называется *вычислительной системой*. Вычислительная система включает в себя технические средства и программное обеспечение, ориентированные на решение определенной совокупности задач. Существует два способа ориентации. Во-первых, вычислительная система может строиться на основе ЭВМ или вычислительного комплекса общего применения и ориентация системы обеспечивается за счет программных средств – прикладных программ и, возможно, операционной системы. Во-вторых, ориентация на заданный класс задач может достигаться за счет использования специализированных ЭВМ и вычислительных комплексов. В этом случае удается при умеренных затратах оборудования добиться высокой производительности. Специализированные вычислительные системы наиболее широко используются при решении задач векторной и матричной, алгебры, а также связанных с интегрированием дифференциальных уравнений, обработкой изображений, распознаванием образов и т. д.

Вычислительные системы, построенные на основе специализированных комплексов, начали интенсивно разрабатываться с конца 60-х годов. В таких системах использовались процессоры со специализированными системами команд, конфигурация комплексов жестко ориентировалась на конкретный класс задач. В последнее десятилетие начались исследования и разработки адаптивных вычислительных систем, гибко приспосабливающихся к решаемым задачам. Адаптация вычислительной системы с целью приспособления ее к структуре реализуемого алгоритма достигается за счет изменения конфигурации системы. При этом соединения между процессорами, а также модулями памяти и периферийными устройствами устанавливаются динамически в соответствии с потребностями задач, обрабатываемых системой в текущий момент времени. В связи с этим адаптивные вычислительные системы иначе называются системами с динамической структурой. За счет адаптации достигается высокая производительность в широком классе задач и обеспечивается устойчивость системы к отказам. Поэтому адаптивные системы рассматриваются как одно из перспективных направлений развития систем обработки данных.

Системы телеобработки. Уже первоначальное применение СОД для управления производством, транспортом и материально-техническим снабжением показало, что эффективность систем можно значительно повысить, если обеспечить ввод данных в систему непосредственно с мест их появления и выдачу результатов обработки к местам их использования. Для этого необходимо связать СОД и рабочие места пользователей с помощью каналов связи. Системы, предназначенные для обработки данных, передаваемых по каналам связи, называются *системами телеобработки данных*.

Состав технических средств системы телеобработки данных укрупненно представлен на рис. 1.3. Пользователи (абоненты) взаимодействуют с системой посредством терминалов (абонентских пунктов), подключаемых через каналы связи к средствам обработки данных – ЭВМ или вычислительному комплексу. Данные передаются по каналам связи в форме сообщений – блоков данных, несущих в себе кроме собственно данных служебную информацию, необходимую для управления процессами передачи и защиты данных от искажений. Программное обеспечение систем телеобработки содержит специальные средства, необходимые для управления техническими средствами, установления связи между ЭВМ и абонентами, передачи данных между ними и организации взаимодействия пользователей с программами обработки данных.

Телеобработка данных значительно повышает оперативность информационного обслуживания пользователей и наряду с этим позволяет создавать крупномасштабные системы, обеспечивающие доступ широкого круга пользователей к данным и процедурам их обработки.

Вычислительные сети. С ростом масштабов применения электронной вычислительной техники в научных исследованиях, проектно-конструкторских работах, управлении производством и транспортом и прочих областях стала очевидна необходимость объединения СОД, обслуживающих отдельные предприятия и коллективы. Объединение разрозненных СОД обеспечивает доступ к данным и процедурам их обработки для всех пользователей, связанных общей сферой деятельности. Так, экспериментальные данные, полученные группой исследователей, могут использоваться при проектно-конструкторских работах, результаты проектирования – при технологической подготовке производства, результаты испытаний и эксплуатации изделий – для совершенствования конструкций и технологии производства и т. д. Чтобы объединить территориально разрозненные СОД в единый комплекс, необходимо, во-первых, обеспечить возможность обмена данными между СОД, связав соответствующие ЭВМ в вычислительные комплексы каналами передачи данных, и, во-вторых, оснастить системы программными средствами, позволяющими пользователям одной системы обращаться к информационным, программным и техническим ресурсам других систем.

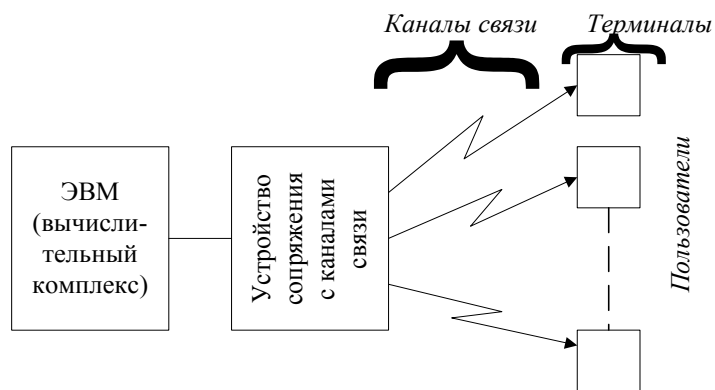


Рис. 1.3. Система телеобработки данных

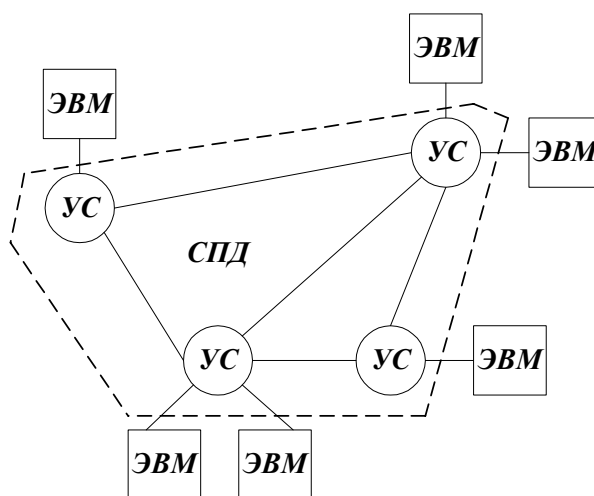


Рис. 1.4. Вычислительная сеть

В конце 60-х годов был предложен способ построения вычислительных сетей, объединяющих ЭВМ (вычислительные комплексы) с помощью базовой сети передачи данных. Структура вычислительной сети в общих чертах представлена на рис. 1.4. Ядром является *базовая сеть передачи данных* (СПД), которая состоит из каналов и узлов связи (УС). Узлы связи принимают данные и передают их в направлении, обеспечивающем доставку данных абоненту. ЭВМ подключаются к узлам базовой сети передачи данных, чем обеспечивается возможность обмена данными между любыми парами ЭВМ. Совокупность ЭВМ, объединенных сетью передачи данных, образует *сеть ЭВМ*. К ЭВМ непосредственно или с помощью каналов связи подключаются терминалы, через которые пользователи взаимодействуют с сетью. Совокупность терминалов и средств связи, используемых для подключения терминалов к ЭВМ, образует *терминальную сеть*. Таким образом, вычислительная сеть представляет собой композицию базовой сети передачи данных, сети ЭВМ и терминальной сети. Такая вычислительная сеть называется глобальной или распределенной (в дальнейшем – «вычислительная сеть», в отличие от локальной). Вычислительные сети используются для объединения ЭВМ, находящихся на значительном расстоянии друг от друга в пределах региона, страны или континента.

В вычислительной сети все ЭВМ оснащаются специальными программными средствами для сетевой обработки данных. На сетевое программное обеспечение возлагается широкий комплекс функций: управление аппаратурой сопряжения и каналами связи: установление соединений между взаимодействующими процессами и ЭВМ; управление процессами передачи данных; ввод и выполнение заданий от удаленных терминалов; доступ программ к наборам данных, размещенных в удаленных ЭВМ, и др. К сетевому программному обеспечению предъявляются следующие требования: сохранение работоспособности сети при изменении ее структуры вследствие выхода из строя отдельных ЭВМ, каналов и узлов связи, а также возможность работы ЭВМ с терминалами различных типов и взаимодействия разнотипных ЭВМ. Функции, возлагаемые на сетевое программное обеспечение, отличаются высоким уровнем сложности и реализуются с использованием специально разработанных методов управления процессами передачи и обработки данных.

Вычислительные сети – наиболее эффективный способ построения крупномасштабных СОД. Использование вычислительных сетей позволяет автоматизировать управление отраслями производства, транспортом и материально-техническим снабжением в масштабе крупных регионов и страны в целом. За счет концентрации в сети больших объемов данных и общедоступности средств обработки значительно улучшается информационное обслуживание научных исследований, повышается производительность труда инженерно-технических работников и качество административно-управленческой деятельности. Кроме того, объединение ЭВМ в вычислительные сети позволяет существенно повысить эффективность их использования. Как показывает практика, стоимость обработки данных в вычислительных сетях, по крайней мере, в полтора раза меньше, чем при использовании автономных ЭВМ.

Локальные вычислительные сети. К концу 70-х годов в сфере обработки данных широкое распространение наряду с ЭВМ общего назначения получили мини и микро-ЭВМ и начали применяться персональные ЭВМ. При этом для обработки данных в рамках одного предприятия или его подразделения использовалось большое число ЭВМ, каждая из которых обслуживала небольшую группу пользователей, а микро-ЭВМ и персональные ЭВМ – отдельных пользователей. В то же время коллективный характер труда требовал оперативного обмена данными между пользователями, т. е. объединения ЭВМ в единый комплекс. В конце 70-х годов разработан эффективный способ объединения ЭВМ, расположенных на незначительном расстоянии друг от друга – в пределах одного здания или группы соседних зданий, с помощью моноканала, (последовательного интерфейса) – локальные вычислительные сети.

Локальная вычислительная сеть (ЛВС) – совокупность близкорасположенных ЭВМ, которые связаны последовательными интерфейсами и оснащены программными средствами,

обеспечивающими информационное взаимодействие между процессами в разных ЭВМ. Типичная структура ЛВС изображена на рис. 1.5. Сопрягаются ЭВМ с помощью моноканала – единого для всех ЭВМ сети канала передачи данных. В моноканале наиболее широко используются скрученная пара проводов, коаксиальный кабель или волоконно-оптическая линия. Длина моноканала не превышает обычно нескольких сотен метров. При этом пропускная способность моноканала составляет 10^5 – 10^7 бит/с, что достаточно для обеспечения информационной связи между десятками ЭВМ. ЭВМ сопрягаются с моноканалом с помощью сетевых адаптеров (СА), иначе контроллеров, реализующих операции ввода – вывода данных через моноканал. Наличие в сети единственного канала для обмена данными между ЭВМ существенно упрощает процедуры установления соединений и обмена данными между ЭВМ. Поэтому сетевое программное обеспечение ЭВМ оказывается более простым, чем в вычислительных сетях, содержащих сеть передачи данных, и легко встраивается даже в микро-ЭВМ. Вследствие этого локальные вычислительные сети оказываются эффективным средством построения сложных СОД на основе микро- и мини-ЭВМ.

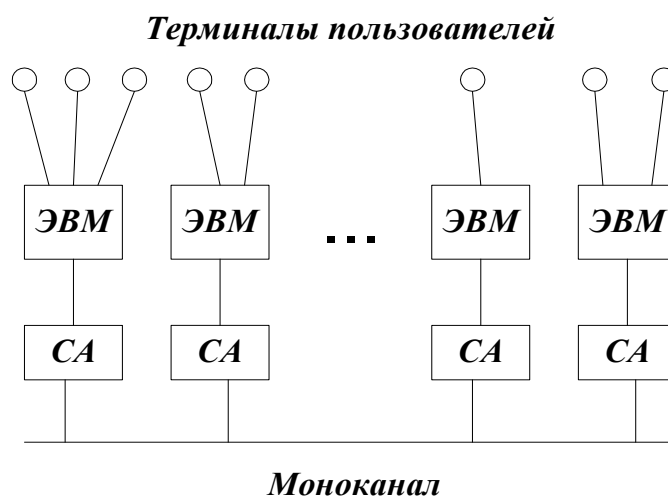


Рис. 1.5. Локальная вычислительная сеть Терминалы пользователей

Локальные вычислительные сети получают широкое применение в системах автоматизации проектирования и технологической подготовки производства, системах управления производством, транспортом, снабжением и сбытом (учрежденческих системах), а также и системах автоматического управления технологическим оборудованием, создаваемых на основе микро- и мини-ЭВМ, в частности в гибких производственных системах.

Классификация СОД. Классифицируются СОД в зависимости от способа построения (рис. 1.6). СОД, построенные на основе отдельных ЭВМ, вычислительных комплексов и систем, образуют класс *сосредоточенных (централизованных) систем*, в которых вся обработка реализуется ЭВМ, вычислительным комплексом или специализированной системой. Системы телеобработки и вычислительные сети относятся к классу *распределенных систем*, в которых процессы обработки данных рассредоточены по многим компонентам. При этом системы телеобработки считаются распределенными в некоторой степени условно, поскольку основные функции обработки данных здесь реализуются централизованно – в одной ЭВМ или вычислительном комплексе.

Существенное влияние на организацию СОД оказывают технические возможности средств, используемых для сопряжения (комплексирования) ЭВМ. Основным элементом сопряжения является *интерфейс*, определяющий число линий для передачи сигналов и данных и способ (алгоритм) передачи информации 110 линиям. Все интерфейсы, используемые в вычислительной технике и связи, разделяются на три класса: параллельные, последовательные и связные (рис. 1.7). *Параллельный интерфейс* состоит из большого числа линий, данные по которым передаются в

параллельном коде – обычно в виде 8 – 128-разрядных слов. Параллельные интерфейсы имеют большую пропускную способность, как правило, 10^6 – 10^8 бит/с. Столь большие скорости передачи данных обеспечиваются за счет ограниченной длины интерфейса, которая обычно составляет от нескольких метров до десятков метров и в редких случаях достигает сотни. *Последовательный интерфейс* состоит, как правило, из одной линия, данные по которой передаются в последовательной ходе. Пропускная способность последовательных интерфейсов обычно составляет 10^5 – 10^7 бит/с при длине линии от десятков метров до километра. *Связные интерфейсы* содержат каналы связи, работа которых обеспечивается аппаратурой передачи данных, повышающей (в основном с помощью физических методов) достоверность передачи. Связные интерфейсы обеспечивают передачу данных на любые расстояния, однако с небольшой скоростью – в пределах от 10^3 до 10^5 бит/с. Применение связных интерфейсов экономически оправдывается на расстояниях, не меньших километра.

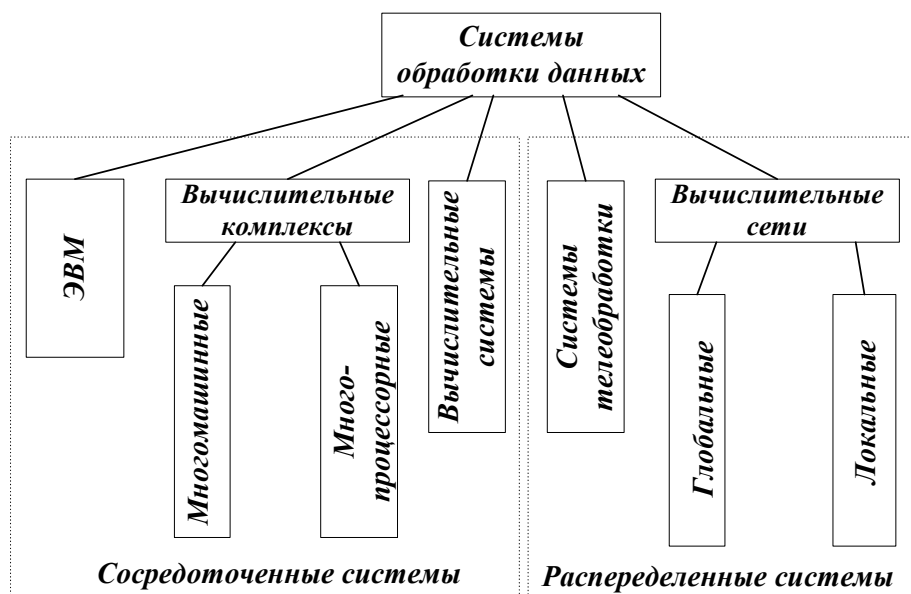


Рис. 1.6. Классификация СОД

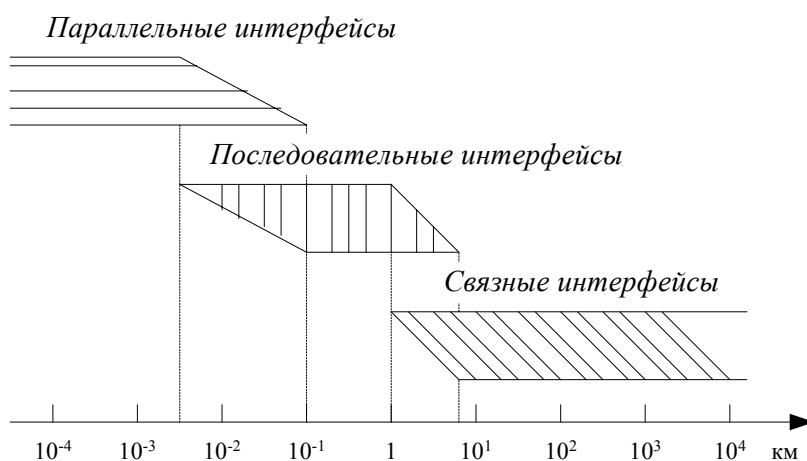


Рис. 1.7. Характеристики интерфейсов

В сосредоточенных системах применяются в основном параллельные интерфейсы, используемые для сопряжения устройств и построения многомашинных и многопроцессорных комплексов, и только в отдельных случаях, чаще для подключения периферийных устройств, применяются последовательные интерфейсы. Параллельные интерфейсы обеспечивают передачу сигналов прерывания, отдельных слов и блоков данных между сопрягаемыми ЭВМ и устройствами.

В распределенных системах из-за значительности расстояний между компонентами применяются последовательные и связанные интерфейсы, которые исключают возможность передачи сигналов прерывания между сопрягаемыми устройствами и требуют представления данных в виде сообщений, передаваемых с помощью операций ввода – вывода. Различие способов предъявления данных в параллельных, последовательных и связанных интерфейсах и в пропускной способности интерфейсов существенно влияет на организацию обработки данных и, следовательно, программного обеспечения СОД.

1.2. СОСТАВ И ФУНКЦИОНИРОВАНИЕ

Системы обработки данных строятся из технических и программных средств, существенно различающихся по своей природе. Поэтому СОД принято рассматривать как совокупность двух составляющих: технических средств и программного обеспечения. Функционирование СОД определяется взаимодействием программных и технических средств, в результате чего свойства системы проявляются как совокупные свойства технических и программных средств.

Технические средства. Основу СОД составляют технические средства – оборудование, предназначенное для ввода, хранения, преобразования и вывода данных. Состав технических средств определяется *структурой (конфигурацией)* СОД, т. е. тем, из каких частей (элементов) состоит система и каким образом эти части связаны между собой. Математическая форма представления структуры – *граф*, вершины которого соответствуют элементам системы, а ребра (дуги) – связи между элементами. Инженерная форма представления структуры – *схема*. Таким образом, схема и граф тождественны по содержанию и различны по форме. В схеме для изображения элементов используются различные геометрические фигуры, а для изображения связей – линии многих типов. За счет этого схема приобретает большую по сравнению с графом наглядность. Основные элементы структуры СОД – устройства: процессоры, устройства запоминающие ввода – вывода, сопряжения с объектами и др. Устройства, связываются с помощью интерфейсов, включающих в себя совокупность линий или каналов передачи данных (линий связи).

Пример структуры, представленной на уровне устройств, приведен на рис. 1.8. В состав рассматриваемого комплекса входят две ЭВМ, каждая из которых снабжена тремя каналами ввода – вывода *МКО*, *СК1* и *СК2*, двумя накопителями на магнитных дисках *НМД1* и *НМД2* и дисплеями *Д1* и *Д2*, подключенными через контроллер *КД* к мультиплексному каналу *МКО*. Машины связаны с общим для них набором внешних запоминающих устройств – накопителями на магнитных дисках *НМД3* и *НМД4* и магнитных лентах *НМЛ1* – *НМЛ4*, которые подключены к селекторным каналам *СК2* через соответствующие контроллеры *КНМД* и *КНМЛ*. К ЭВМ подключены мультиплексоры передачи данных *МПД1* и *МПД2*, каждый из которых обслуживает четыре канала связи *КС1* – *КС4* и *КС5* – *КС8*. На рисунке линиями представлены следующие интерфейсы: интерфейс прямого управления, сопрягающий процессоры *ЭВМ1* и *ЭВМ2*; интерфейсы оперативной памяти, связывающее оперативную память с процессором и каналами ввода – вывода *МКО*, *СК1* в *СК2*; интерфейсы ввода – вывода, связывающие каналы ввода – вывода с контроллерами запоминающих устройств и устройств ввода – вывода; малые интерфейсы, посредством которых накопители и устройства ввода – вывода подключаются к соответствующим контроллерам.

Структура сложных систем при представлении ее на уровне устройств может оказаться настолько сложной, что теряет обзримость и выходит за рамки возможностей методов исследования, используемых при анализе и синтезе систем. В таких случаях структура описывается на более высоком уровне, когда в качестве элементов выступают ЭВМ, многопроцессорные комплексы и сложные подсистемы, которые изображаются одной вершиной графа. Таким образом, элемент структуры СОД – это прежде всего удобное понятие, но не физическое свойство объекта. Главное требование к изображению структуры – информативность.

Структура СОД дает общее представление о составе технических средств и связей между ними. Дополнительные сведения о технических средствах даются в форме спецификации, где для каждого элемента структуры и каждого типа связей между элементами указывается: наименование элемента, приведенное на структурной схеме; тип устройства, соответствующего элементу структурной схемы; технические характеристики устройства или средства связи (производительность, емкость памяти, пропускная способность).

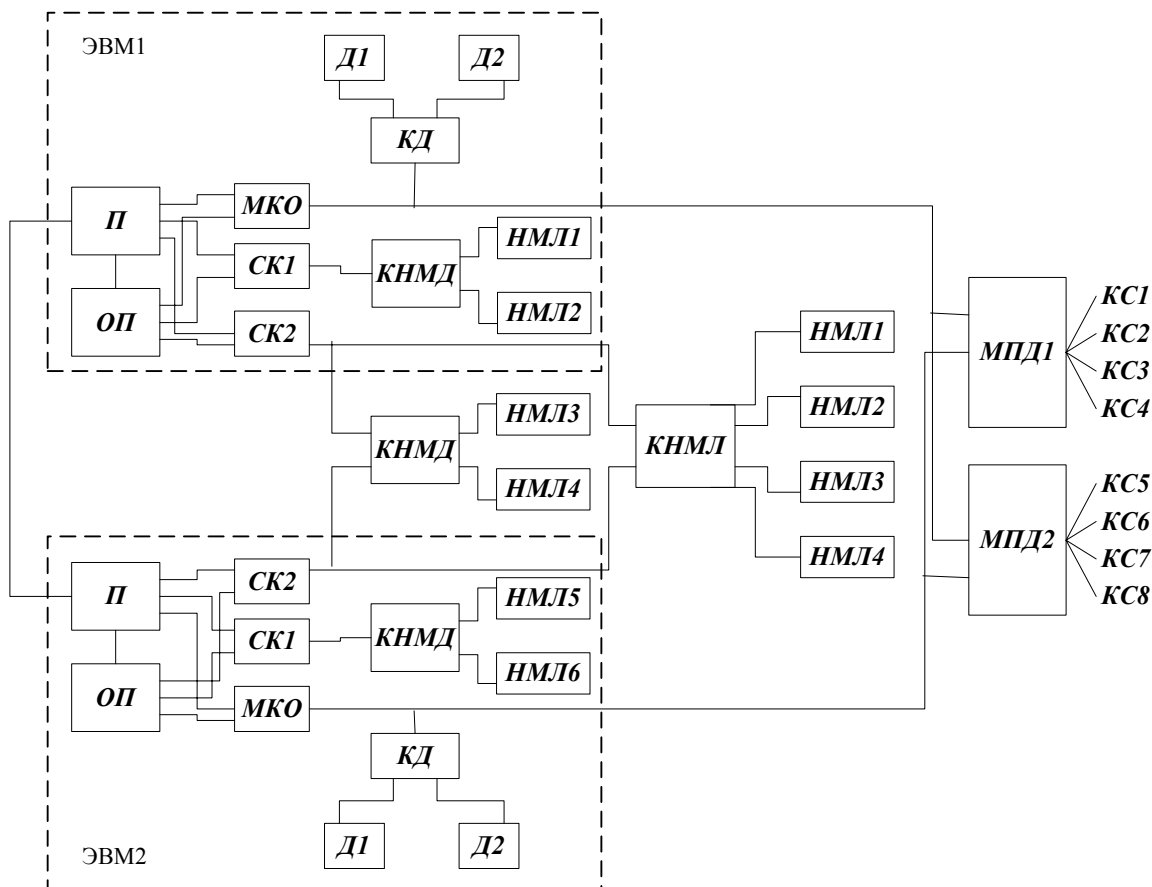


Рис. 1.8. Двухмашинный вычислительный комплекс. П – процессор; ОП – оперативная память; МК – мультиплексный канал; КНМД – контроллер НМД; КНМЛ – контроллер НМЛ; КД – контроллер дисплеев; МПД – мультиплексор передачи данных; КС – канал связи; Д – дисплей

В связи с процессами обработки данных технические средства рассматриваются как совокупность ресурсов двух типов: устройств и памяти. Устройство – ресурс, используемый для преобразования и ввода – вывода данных, разделяемый между процессами (задачами) во времени. Примеры устройств – процессоры, каналы ввода – вывода, периферийные устройства (ввода – вывода и внешние запоминающие) и каналы передачи данных. В каждый момент времени устройство используется одним процессом, реализуя соответствующие операции: преобразование или ввод – вывод данных. Основная характеристика устройства – производительность, определяемая числом операций, выполняемых в секунду, или пропускная способность определяемая количеством единиц информации (байтов), передаваемых в секунду. Память – ресурс, используемый для хранения данных и разделяемый между процессами по объему к времени. Примеры – оперативная память и накопители на магнитных дисках. Основная характеристика памяти – емкость, определяемая предельным количеством информации, размещаемой в памяти. В одной памяти одновременно могут размещаться данные, относящиеся к нескольким процессам. Накопитель на магнитных дисках содержит два ресурса, являясь одновременно памятью определенной емкости и устройством, обслуживающим операции ввода – вывода данных.

Таким образом, состав технических средств определяет номенклатуру ресурсов, используемых для хранения, ввода – вывода и преобразования данных. Конфигурация связей между устройствами определяет пути передачи данных в системе и порядок доступа процессов к устройствам и данным, хранимым в памяти.

Программное обеспечение. Технические средства СОД реализуют элементарные операции ввода – выводе и обработки данных. Требуемый набор функций, определяемых назначением СОД, обеспечивается совокупностью программ – программным обеспечением СОД.

Программное обеспечение СОД строится по многоуровневому, иерархическому, принципу. Основные процессы обработки данных описываются в терминах операций над математическими и логическими элементами данных, вводимых проблемно и процедурно-ориентированными языками программирования. Эти операции с помощью программных средств более низких уровней интерпретируются как более простые операции, в конце концов, сводятся к операциям, реализуемым техническими средствами СОД.



Рис. 1.9. Многоуровневая организация СОД

Пример многоуровневой реализации функций в СОД приведен на рис. 1.9. Технические средства СОД обеспечивают реализацию элементарных функций – операций ввода, хранения, преобразования и вывода данных, которые выполняются с помощью схем и средств микропрограммного управления. Функции, реализуемые техническими средствами, относятся к первому, низшему, уровню иерархии. Функции более высоких уровней сложности обеспечиваются программным обеспечением СОД, включающим в себя операционную систему и прикладное программное обеспечение.

Операционная система (ОС) – совокупность программ, предназначенных для управления работой СОД и реализации наиболее массовых процедур взаимодействия с пользователями, ввода – вывода, хранения и преобразования данных. Управление работой СОД сводится к управлению процессами и ресурсами, обеспечивающему эффективное использование оборудования СОД и требуемое качество обслуживания пользователей. Функции управления работой СОД реализуются *управляющими программами* ОС, включающими в свой состав супервизор, программы управления заданиями и данными. *Супервизор* контролирует состояние всех технических средств и процессов (задач) и управляет ими, обеспечивая необходимый режим обработки данных, путем распределения процессов в пространстве и времени. Супервизор выделяет задачам области (разделы)

памяти и устройства ввода – вывода, инициирует выполнение процессором программы, начинает операции ввода – вывода и обрабатывает сигналы прерывания, отмечающие окончание операций ввода – вывода и особые ситуации, возникающие при выполнении программ и работе устройств.

Программы управления заданиями обеспечивают ввод и интерпретацию команд операторов, управляющих работой СОД, и заданий, формируемых пользователями СОД. Операторы с помощью специальных команд воздействуют на порядок функционирования и получают информацию о текущем состоянии СОД. Эти программы интерпретируют задания в виде соответствующих действий и обеспечивают их необходимыми ресурсами – разделами оперативной и внешней памяти, устройствами ввода – вывода, наборами данных и др. Задания, обеспеченные ресурсами, необходимыми для их выполнения, образуют задачи. Управление задачами реализуется супервизором. Для обращения к программам управления заданиями применяется язык управления заданиями, в терминах которого пользователи и операторы, управляющие работой системы, записывают задания на выполнение работ в системе.

Программы управления данными обеспечивают доступ к наборам данных и организацию работы устройств ввода – вывода. Средства управления данными настраивают программы на работу с конкретными наборами данных и устройствами, в которых хранятся наборы, и за счет этого создают возможность при программировании задач манипулировать с данными как с логическими объектами, не связанными с конкретными устройствами. Таким образом, управление данными сводится к сопряжению программ с наборами данных и устройствами, а использование этих устройств контролируется и координируется супервизором. Функции, реализуемые управляющими программами ОС, относятся ко второму и третьему уровню функций системы (см. рис. 1.9).

Функции ОС расширяются за счёт средств *системного программного обеспечения* – программных средств телеобработки, управления базами данных, сетевой обработки и др. Системное программное обеспечение является основой для построения прикладного программного обеспечения и предоставляет пользователю средства, необходимые для работы со специальными устройствами (например, с аппаратурой передачи данных и удаленными терминалами) или для специальной обработки данных. Функции, реализуемые средствами системного программного обеспечения, относятся к четвертому уровню иерархии.

К пятому уровню относятся функции, выполняемые *системными обрабатывающими программами* ОС. Эти программы включают в себя: трансляторы с языков программирования: редакторы связей, обеспечивающие сборку программных модулей в программы с заданной структурой; средства отладки программ и перемещения наборов данных с одних носителей на другие и т. д. Функции, обеспечиваемые трансляторами; представляются в виде языков программирования: машинно-, процедурно- и проблемно-ориентированных языков, языков генерации программ ввода – вывода и др.

Прикладное программное обеспечение – совокупность прикладных программ, реализующих функции обработки данных связанные с конкретной областью применения системы. В системах автоматизации проектирования радиоэлектронной аппаратуры прикладные программы обеспечивают анализ электронных схем, размещение электронных элементов по конструктивным единицам, разводку соединений на печатных платах и т. д.: в автоматизированных системах управления производством – календарное и оперативное планирование производства на предприятии и а низовых производственных подразделениях, учет и анализ производственной деятельности и т. д. Состав прикладных программ определяется назначением системы.

К программным средствам СОД примыкают наборы данных, рассматриваемых как особая составляющая – информационное обеспечение СОД. Наборы данных – совокупность логически связанных элементов данных, организованных по определенным правилам и снабженных описанием, доступным системе программирования (средствам управления

данными). Наборы данных снабжаются именами, с помощью которых программы обращаются к соответствующим наборам и их элементам. Одни и те же наборы данных могут использоваться многими прикладными программами. Чтобы исключить необходимость представления одних и тех же данных в различной форме, вариантах и сочетаниях, ориентированных на разные программы, необходимо обеспечить независимость данных и программ. Это достигается за счет организации данных в виде специальных структур – баз и банков данных, а также использования совокупности программных средств, предназначенных для выборки, модификации и добавления данных, – *системы управления базами данных*. Организация данных в форме баз обеспечивает независимость прикладных программ от логической и физической организации базы данных, в результате чего изменения в программах не влекут за собой изменения базы и реорганизацию базы данных, не требует внесения изменений в программы, оперирующие с данными.

Функционирование СОД. Функционирование СОД представляется в виде процессов. *Процесс*¹ – это динамический объект, реализующий целенаправленный акт обработки данных. Процессы разделяются на прикладные и системные. *Прикладные процессы* реализуют основные функции СОД, заданные прикладными программами или обрабатывающими программами ОС, а инициируются заданиями пользователей или сигналами, поступающими в СОД из внешней среды. Примеры прикладных процессов: решение прикладной задачи; редактирование, трансляция и сборка программы; сортировка набора данных и др. *Системные процессы* реализуют вспомогательные функции, обеспечивающие работу СОД. Примеры системных процессов: системный ввод; системный вывод; перемещение страниц, а виртуальной памяти; работа супервизора и др. Как правило, системные процессы существуют в течение всего периода работы СОД – от момента включения до момента выключения СОД.

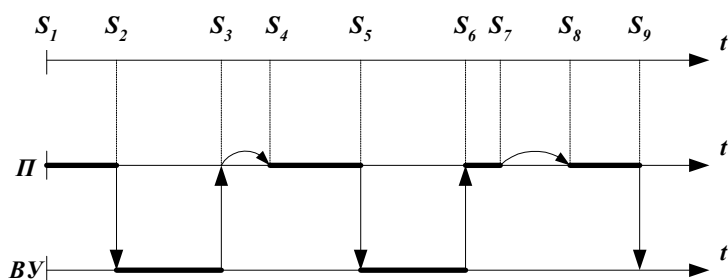


Рис. 1.10. Временная диаграмма вычислительного процесса

Процесс P_i описывается тройкой $P_i = \langle t_i, A_i, T_i \rangle$, где t_i – момент инициирования процесса, A_i – атрибуты процесса, определяющие имена источника, инициировавшего процесс, пользователя, задания, режим обработки данных, приоритет процесса и др., и T_i – трасса процесса. *Трасса процесса* – последовательность событий, связанных с изменением состояния процесса. Трасса процесса представляется в виде упорядоченного множества событий $T_i = \{S_1, S_2, \dots, S_M\}$, имевших место в моменты времени t_1, t_2, \dots, t_M , причем $t_1 \leq t_2 \leq \dots \leq t_M$. К событиям относятся моменты ввода задания, начала и завершения обработки шагов (пунктов) задания, начала и окончания выполнения процессов в устройствах СОД, начала использования и освобождения разделов памяти, предоставляемых процессу в запоминающих устройствах, и др. Каждое событие связывается с моментом его возникновения, программой, реализующей процесс, и ресурсом, обслуживающим процесс. Таким образом, трасса характеризует динамику процесса – развитие процесса во времени и пространстве. Трасса может быть представлена в виде временной диаграммы, на рис. 1.10 изображающей выполнение программы процессором и внешним устройством. Отрезки,

¹ В ряде публикаций – вычислительный процесс.

выделенные на осях жирными линиями, соответствуют периодам, когда процессор и внешнее устройство заняты выполнением программы. Дугами обозначены интервалы времени, в течение которых процесс находится в состоянии ожидания, т. е. не обслуживается ни одним устройством, ожидая момента освобождения устройства.

Таким образом, функционирование СОД выражается в форме процессов выполнения программ. Процесс выполнения программы связан с использованием ресурсов СОД, а также наборов данных и самих программ. Следовательно, характерной чертой процесса является его одновременная связь и с выполнением программ и с работой технических средств СОД.

Рабочая нагрузка. Процесс функционирования СОД существенно зависит от состава заданий, исходных данных и сигналов, поступающих на вход СОД. Весь объем поступающей информации принято называть рабочей нагрузкой СОД. При проектировании к эксплуатации СОД наибольший интерес представляет потребность заданий в ресурсах: оперативной и внешней памяти, процессорном времени, устройствах ввода – вывода и др. Поэтому рабочую нагрузку, относящуюся к промежутку времени T , определяют в виде множества характеристик заданий.

$$L = \{l_i\} = \{ \langle A_i, \Theta_{i1}, \dots, \Theta_{iN} \rangle \}, \quad (1.1)$$

где l_i – описание 1-го задания, устанавливающее его атрибуты A_i и потребность задания $\Theta_{i1}, \dots, \Theta_{iN}$ в ресурсах $1, \dots, N$. Например, значение Θ_{i1} может определять емкость области оперативной памяти, необходимой заданию, Θ_{i2} – число выполняемых процессором операций, Θ_{i3} – количество вводимых данных и т. д.

Число заданий, обрабатываемых СОД за промежутки времени, дающий полное представление о рабочей нагрузке, обычно весьма велико. Поэтому описание рабочей нагрузки в виде (1.1) оказывается, как правило, громоздким. Для представления рабочей нагрузки в компактной форме потребность заданий в ресурсах характеризуется среднестатистическими значениями объема ресурсов, приходящимися на одно задание.

Рабочая нагрузка зависит от назначения (сферы применения) СОД и оказывается различной для систем, оперирующих с разными классами задач: инженерно-техническими, планоно-экономическими, учетно-статистическими и др.

1.3. ХАРАКТЕРИСТИКИ И ПАРАМЕТРЫ

При проектировании СОД, стремятся обеспечить наиболее полное соответствие системы своему назначению. Степень соответствия системы своему назначению называется *эффективностью (качеством) системы*. Для сложных систем, какими являются СОД, эффективность не удастся определить одной величиной, и поэтому ее представляют набором величин, называемых характеристиками системы². Набор характеристик формируется таким образом, чтобы в своей совокупности они давали наиболее полное представление об эффективности системы. Основными характеристиками СОД являются производительность, время ответа, надежность и стоимость. В дополнение к ним используются следующие характеристики: габариты, масса, потребляемая мощность, диапазон рабочих температур, ремонтпригодность и др.

Характеристики зависят от организации системы – структуры, состав» программного обеспечения, режима функционирования системы и др. Применительно к задачам оценки эффективности организация СОД определяется в виде математических объектов, называемых параметрами системы. В качестве параметров используются величины, определяющие, например, число и быстродействие устройств, емкость памяти, рабочую нагрузку и др. Наряду с этими величинами в качестве параметров могут использоваться,

² Характеристики иначе называются частными показателями эффективности (качества) системы. Однако характеристика связывается с определённым свойством системы, а показатель – с величиной и способом ее вычисления, т. е. характеристика – понятие более, высокого ранга, чем показатель.

такие математические объекты, как множества, графы, алгоритмы и др. В число параметров включаются все объекты, характеризующие первичные аспекты организации системы и существенно влияющие на характеристики.

Таким образом, характеристики определяют свойства системы как целого, проявляющиеся в процессе эксплуатации системы и зависящие от ее организации, представляемой соответствующим набором параметров. В математическом аспекте характеристики можно рассматривать как наименования функций, аргументами которых являются параметры.

Рассмотрим способы оценки основных характеристик СОД и наборы параметров, влияющих на характеристики.

Производительность. Производительность – характеристика вычислительной мощности системы, определяющая количество вычислительной работы, выполняемой системой за единицу времени. В настоящее время отсутствует общепринятая методика оценки производительности СОД, что связано в первую очередь с отсутствием единиц для измерения количества вычислительной работы. Поэтому для оценки производительности используется широкая номенклатура величин – показателей производительности, которые и в отдельности и в совокупности не удовлетворяют в полной мере потребностям теории и практики проектирования и эксплуатации. СОД. Ниже определяются основные способы оценки производительности³.

Технические средства СОД (ЭВМ и периферийное оборудование) обладают производительностью вне связи с операционной системой, прикладным программным обеспечением и режимом эксплуатации системы. Производительность технических средств оценивается их быстродействием – числом операций, выполняемых ЭВМ и устройствами за секунду. Совокупность значений $V = (V_1, \dots, V_N)$, определяющих быстродействия устройств $1, \dots, N$, входящих в состав системы, характеризует номинальную производительность система. Чтобы оценка номинальной производительности была по возможности простой, стремятся уменьшить число составляющих в наборе (V_1, \dots, V_N) . Это достигается двумя способами. Во-первых, быстродействие устройств, выполняющих одинаковые операции и способных работать параллельно, представляют суммарным быстродействием. За счет этого номинальное быстродействие может быть представлено как набор значений, а именно суммарное быстродействие процессоров, внешней памяти, средств ввода и вывода, например: быстродействие процессоров 1,2 млн. операций в секунду; быстродействие внешней памяти 200 обращений в секунду; скорость ввода 6 тыс. символов в секунду. Во-вторых, быстродействие внешней памяти и подсистем ввода – вывода может считаться несущественным и тогда номинальную производительность характеризуют одним значением – суммарным быстродействием процессоров системы.

Номинальная производительность характеризует только потенциальные возможности устройств, которые не могут быть использованы полностью. Этому препятствует влияние структуры связей между устройствами на их производительность, что проявляется в изменении скорости работы одних устройств при работе других. Так, из-за того, что процессор и каналы ввода – вывода подключены к общей оперативной памяти, увеличение скорости ввода – вывода приводит к уменьшению производительности процессора; суммарная производительность устройств ввода – вывода, подключенных к мультиплексному каналу, ограничена пропускной способностью канала, фактическая производительность накопителей, подключенных к блок-мультиплексному каналу, меньше их суммарного номинального быстродействия и т. д. Чтобы оценить влияние первой группы факторов – структуры системы на быстродействие устройств, используется специальная характеристика – комплексная производительность. Комплексная производительность оценивается набором быстродействий устройств V_1^*, \dots, V_N^* , обеспечиваемых при совместной их работе, т. е. в составе комплекса технических средств. По вышеописанным причинам комплексная

³ Подробный анализ показателей производительности приведен в работе [18].

производительность ниже номинальной: $V_1^* \leq V_1, \dots, V_N^* \leq V_N$. Способ оценки комплексной производительности до сих пор не определен. Один из подходов к ее оценке сводится к следующему. Некоторым образом определяется типовая смесь операций ввода, обращения к внешней памяти, обработки и вывода данных, на основе которой создается синтетическая искусственная программа, порождающая процесс с заданной смесью операций. Путем прогона синтетической программы и измерения времени ее выполнения оценивается комплексная производительность системы.

Показателем использования устройства в процессе работы системы является загрузка. Загрузка i -го устройства определяется отношением T_i/T , где T_i – время, в течение которого устройство работало, и T – продолжительность работы системы. В течение промежутка времени $T - T_i$ устройство простаивает. Очевидно, что загрузка $\rho_i \leq 1$. Если загрузка устройств $1, \dots, N$ равна ρ_1, \dots, ρ_N , соответственно, то количество работы, выполняемой устройствами с быстродействием V_1, \dots, V_N за единицу времени, равно $\rho_1 V_1, \dots, \rho_N V_N$. Совокупность значений $\rho_1 V_1, \dots, \rho_N V_N$ характеризует производительность технических средств с учетом простоев, возникающих в процессе функционирования системы. Таким образом, оценка фактической производительности системы сводится к оценке загрузки устройств в конкретных условиях работы системы.

На загрузку устройств существенно влияет режим обработки задач, реализуемый управляющими программами операционной системы. Влияние операционной системы проявляется, например, в следующем. Организация системного ввода и вывода связана с использованием процессора и внешних запоминающих устройств для промежуточного хранения вводимых и выводимых наборов данных. В результате этого часть времени процессора, каналов ввода – вывода и внешних запоминающих устройств тратится на обслуживание ввода – вывода. Такая же ситуация возникает при организации в системе виртуальной памяти, режима разделения времени и обеспечения других вспомогательных функций.

Чтобы оценить влияние операционной системы на производительность технических средств СОД, используется специальная характеристика – системная производительность. *Системная производительность* определяется набором значений $\rho_1 V_1, \dots, \rho_N V_N$, в котором загрузка ρ_1, \dots, ρ_N определена при совместной работе комплекса технических средств под управлением операционной системы. Из-за вышеописанных факторов системная производительность ниже комплексной ($\rho_1 V_1 \leq V_1^*, \dots, \rho_N V_N \leq V_N^*$) и, следовательно, ниже номинальной ($\rho_1 V_1 \leq V_1, \dots, \rho_N V_N \leq V_N$). В настоящее время общепринятая методика оценки системной производительности отсутствует.

Для СОД, находящихся в эксплуатации или разрабатываемых для конкретного применения, класс задач полностью определен, по крайней мере, статистически, т. е. определена рабочая нагрузка СОД. В таком случае производительность оценивается на рабочей нагрузке и называется *системной производительностью* или кратко – *производительностью*.

Производительность наиболее просто оценивается числом задач, решаемых системой за единицу времени: λ задач/ч. Эта оценка информативна только для конкретной области применения СОД и ни о чем не свидетельствует, если не определен класс решаемых задач. По этой причине она используется, когда анализируются варианты организации одной СОД, и не может применяться для сравнения СОД, работающих с различными наборами задач.

Рассмотрим способы определения производительности на рабочей нагрузке для систем, находящихся в эксплуатации.

Пусть за время T система завершила обработку n задач (заданий). Тогда производительность системы за время T составляет

$$\lambda = n/T \quad (1.2)$$

задач в единицу времени (например, в час).

Обычно задачи поступают на обработку в случайные моменты времени и время пребывания задач в системе зависит от состава смеси (числа и характеристик) задач, одновременно обрабатываемых системой. В результате этого число задач n , обработанных системой за время T , – случайная величина и производительность λ в интервале T оценивается с погрешностью, имеющей статистическую природу и зависящей от случайной величины n и ее дисперсии. С увеличением длительности интервала T значение n возрастает и погрешность оценки λ стремится к нулю при $T \rightarrow \infty$.

Другой способ определения производительности λ через среднее значение интервала между моментами окончания обработки задач. В этом случае в течение времени T регистрируются интервалы между моментами завершения обработки задач τ_1, \dots, τ_n (рис. 1.11).

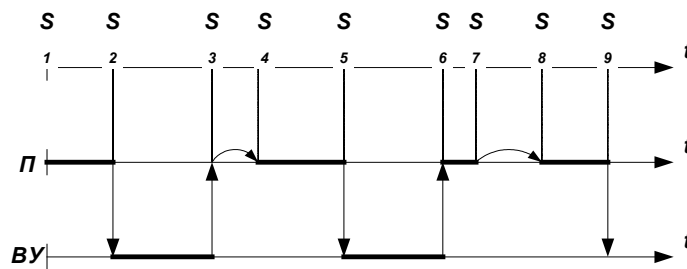


Рис. 1.11. Поток задач на входе и выходе системы.

Среднее значение этого интервала

$$\tau = \frac{1}{n} \sum_{i=1}^n \tau_i$$

определяется интенсивностью выходного потока задач, и производительность системы

$$\lambda = 1/\tau \quad (1.3)$$

Оценки производительности (1.2) и (1.3) совпадают, если начало и конец промежутка времени T совпадают с моментами окончания обработки задач.

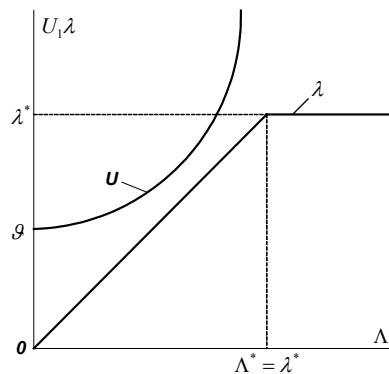


Рис. 1.12. Влияние интенсивности входного потока задач на производительность к время ответа.

Рассмотрим зависимость между двумя величинами: средним числом задач, поступающих в единицу времени на вход системы, – интенсивностью входного потока задач Λ – и средним числом задач, покидающих систему за единицу времени, – интенсивностью выходного потока задач λ . Зависимость представлена на рис. 1.12. В области $0 \leq \Lambda \leq \Lambda^*$ интенсивность выходного потока полностью определяется интенсивностью входного потока: $\lambda = \Lambda$. При $\Lambda > \Lambda^*$ система из-за ограниченности ресурсов – числа и быстродействия устройств, а также емкости памяти – не может в течение единицы времени обслужить все поступившие на обработку задания и интенсивность выходного потока λ , достигнув предельного значения λ^* , остается постоянной при любых значениях $\Lambda > \Lambda^*$, причем $\Lambda^* = \lambda^*$. Значение λ^* определяет максимальную производительность системы для заданного класса задач и является характеристикой самой системы, не зависящей от интенсивности входного потока задач. Таким образом, производительность системы – ограниченная сверху величина: $0 \leq \lambda \leq \lambda^*$. В области $0 \leq \Lambda < \lambda^*$ все ресурсы системы в какой-то степени недоиспользуются. В области $\Lambda \geq \lambda^*$ по крайней мере один ресурс загружен полностью. Остальные ресурсы могут быть недогружены из-за нехватки одного или одновременно нескольких ресурсов.

На производительность наиболее существенно влияют следующие параметры:

- 1) число и быстродействие устройств, емкость оперативной и внешней памяти, с увеличением которых производительность может возрасти, а также структура системы и пропускная способность связей между элементами системы;
- 2) режим обработки задач, определяющий порядок распределения ресурсов системы между задачами, поступающими на обработку;
- 3) рабочая нагрузка, в первую очередь объем вводимых, хранимых в памяти, выводимых данных и число процессорных операций, необходимых для решения задачи.

Оценка производительности в виде числа задач, решаемых системой за единицу времени, имеет смысл только для конкретной задачи, работающей с заданным множеством задач. Чтобы сравнивать производительность различных систем, обрабатывающих различные классы задач, производительность на рабочей нагрузке определяют объемом вычислительной работы, выполняемой системой за единицу времени. Такую оценку представляют набором значений P_1, \dots, P_N , составляющие которого определяют объем обработки, ввода и вывода данных в единицу времени. Например, P_1 – число процессорных операций в секунду, P_2 – число вводимых за секунду символов. Как и при оценке производительности λ числом задач в единицу времени, производительность системы в объеме работ P_1, \dots, P_N характеризуется предельным значением P_1^*, \dots, P_N^* , достигаемым при насыщении системы.

Оценки производительности λ и P_1, \dots, P_N связаны следующим образом. Пусть $\Theta_1, \dots, \Theta_N$ – сложность вычислений, характеризуемая средним числом операций типа 1, ..., N выполняемых при решении одной задачи, причем значения $\Theta_1, \dots, \Theta_N$ включают в себя все операции, в том числе относящиеся к системным процессам. Тогда

$$P_n = \lambda \Theta_n, \quad n = 1, \dots, N \quad (1.4)$$

$$\lambda = P_n / \Theta_n \quad (1.5)$$

Различия в оценках λ , получаемых для различных индексов n , свидетельствует о погрешностях в измерении P_n и Θ_n . Таким образом, оценки производительности λ и P_1, \dots, P_N однозначно связаны через характеристика $\Theta_1, \dots, \Theta_N$ задач.

Пусть известны характеристики задач $\Theta_1, \dots, \Theta_N$ и суммарное быстродействие V_1, \dots, V_N устройств, реализующих операций типа $1, \dots, N$ соответственно. В предположении, что все устройства могут работать параллельно во времени и режим обработки задач, задаваемый управляющими программами операционной системы, обеспечивает параллельную работу устройств, можно получить верхнюю оценку максимальной производительности системы

$$\lambda^* \leq \min(V_1 / \Theta_1, \dots, V_N / \Theta_N) \quad (1.6)$$

Значения V_n / Θ_n определяют максимальную производительность устройств типа $n=1, \dots, N$, характеризуемую числом задач, которые способны обслужить устройства за единицу времени. Наименее производительная в заданном классе задач группа однотипных устройств и определит производительность системы. Оценка (1.6) будет ближе к реальному значению λ^* , если вместо V_1, \dots, V_N подставить значения V_1^*, \dots, V_N^* , характеризующие комплексную производительность системы.

Время ответа. Время ответа, иначе время пребывания заданий, (задач) в системе, – длительность промежутка времени от момента поступления задания в систему до момента окончания его выполнения. На рис. 1.11 указано время ответа u_1, u_2, \dots , для заданий J_1, J_2, \dots соответственно.

В общем случае время ответа – случайная величина, что обусловлено следующими факторами:

- 1) влиянием исходных данных на число операций ввода, обработки и вывода данных и непредсказуемостью значений исходных данных;
- 2) влиянием состава смеси задач, одновременно находящихся в системе, и непредсказуемостью состава смеси из-за случайности момента поступления задач на обработку.

Время ответа как случайная величина наиболее полно характеризуется функцией распределения $P(u < x)$ или функцией плотности вероятностей $p(u)$. Чаще всего время ответа оценивается средним значением, которое определяется как статистическое среднее случайной величины $u_i, i=1, \dots, n$, наблюдаемой для задач J_i :

$$U = \frac{1}{n} \sum_{i=1}^n u_i$$

Время ответа складывается из двух составляющих: времени выполнения задачи и времени ожидания. *Время выполнения* задачи при отсутствии параллельных процессов равно суммарной длительности всех этапов процесса – ввода, обращения к внешней памяти, процессорной обработки и вывода. Время выполнения задачи зависит от сложности вычислений $\Theta_1, \dots, \Theta_N$ и быстродействия V_1, \dots, V_N устройств $1, \dots, N$:

$$g = \sum_{i=1}^N \Theta_i / V_i \quad (1.7)$$

Время ожидания – сумма промежутков времени, в течение которых задача находилась в состоянии ожидания требуемых ресурсов. Ожидание, возникающее при мультипрограммной обработке, когда ресурс, необходимый задаче, занят другой задачей и первая задача не выполняется, ожидая освобождения ресурса. Время ожидания зависит в первую очередь от режима обработки задач и интенсивности входного потока задач (заданий).

Таким образом, время ответа зависит от тех же параметров, что и производительность: структуры и характеристик технических средств, режима обработки и характеристик задач. Зависимость среднего времени ответа U от интенсивности входного потока задач Λ приведена на рис. 1.12. При $\Lambda \rightarrow 0$ время ответа $U \rightarrow \Theta$ где Θ определяется (1.7). С увеличением Λ среднее время ответа монотонно возрастает и может принимать сколь угодно большие значения, если интенсивность входного потока Λ превышает производительность системы λ^* в течение сколь угодно большого периода времени.

Среднее время ответа характеризует быстроту реакций системы на входные воздействия: задания, запросы абонентов и т. п. Качество системы тем выше, чем меньше среднее время ответа.

Характеристики надежности. *Надежность* – свойство системы выполнять возложенные на нее функции в заданных условиях функционирования с заданными показателями качества: достоверностью результатов, пропускной способностью, временем ответа и др. Работоспособность системы или отдельных ее частей нарушается из-за отказов аппаратуры – выхода из строя элементов или соединений.

Важнейшая характеристика надежности – *интенсивность отказов*, определяющая среднее число отказов за единицу времени, как правило, за один час. Интенсивность отказов зависит от числа элементов и соединений, составляющих систему. Если любой отказ носит катастрофический характер, т. е. приводит к нарушению работоспособности системы, то

интенсивность отказов в системе $\lambda_0 = \sum_{i=1}^n \lambda_i$, где λ_i – интенсивность отказов i -го элемента

или соединения и n – число элементов и соединений в системе. Так, если $\lambda_0 = 10^{-2}$ ч, то в среднем за 100 ч происходит один отказ. Средний промежуток времени между двумя смежными отказами называется *средней наработкой на отказ* и равен $T_0 = 1/\lambda_0$. Так, если $\lambda_0 = 10^{-2}$ ч, то наработка на отказ составляет 100 ч. Промежуток времени между отказами – случайные величины со средним значением T_0 , которые, как правило, распределены по экспоненциальному закону. При этом вероятность того, что за время t превзойдет отказ, $P(t < x) = 1 - e^{-t/T_0}$. Так, если $T_0 = 100$ ч, то вероятность того, что в течение 100 ч работы системы произойдет отказ, $P(t < 100) \approx 0,63$ и с вероятностью 37 % отказ произойдет за время большее 100 ч.

Работоспособность системы, нарушенная в результате отказа, восстанавливается путем ремонта системы. Ремонт состоит в выявлении причины нарушений работоспособности – диагностике системы и в восстановлении работоспособности путем замены неисправного элемента. Промежуток времени затрачиваемой на восстановление работоспособности системы, называется временем восстановления. Его длительность зависит от сложности системы, степени совершенства средств диагностики и уровня ремонтпригодности системы. Время восстановления – случайная величина, характеризуемая средним значением T_n – средним временем восстановления.

С учетом средней наработки на отказ T_0 и среднего времени восстановления T_n надежность системы характеризуется коэффициентом готовности

$$K_r = T_0 / (T_0 + T_n) \quad (1.8)$$

определяющим долю времени, в течение которого система работоспособна.

Значение $1 - K_r$ представляет собой долю времени, в течение которого система неработоспособна, ремонтируется. Так, если $K_r = 0,95$, то 95% времени система работоспособна и 5% времени затрачивается на ее ремонт. Кроме того, коэффициент готовности определяет вероятность того, что в произвольный момент времени система

работоспособна, а значение $1 - K_r$ – вероятность того, что в этот момент времени система находится в состоянии восстановления.

Надежность системы может быть повышена за счет резервирования ее элементов – дублирования, троирования и т. д. Однако резервирование приводит к существенному увеличению стоимости системы.

Стоимость. Стоимость СОД – это суммарная стоимость технических средств и программного обеспечения. Стоимость технических средств определяется их составом и техническими характеристиками, Устройства с более высокими техническими характеристиками – быстродействием, емкостью, надежностью – имеют более высокую стоимость. Стоимость программного обеспечения определяется в основном затратами на разработку программ и тиражируемостью программ – числом систем, в которых используются программы. Затраты на разработку программ наиболее существенно зависят от сложности программ.

Стоимость СОД влияет на стоимость решения задачи, которая определяется стоимостью ресурсов используемых задачами:

$$S = \sum_{i=1}^N c_i \Theta_i$$

где c_i – стоимостной коэффициент, определяющий стоимость использования единицы ресурса i (миллиона процессорных операций, килобайта памяти и др.), и Θ_i – объем ресурса, используемый задачами.

1.4. РЕЖИМЫ ОБРАБОТКИ ДАННЫХ

Режим обработки данных – способ выполнения заданий (задач), характеризующийся порядком распределения ресурсов системы между заданиями (задачами). Требуемый режим обработки данных обеспечивается управляющими программами операционной системы, которые выделяют заданиям оперативную и внешнюю память, устройства ввода – вывода, процессорное время и прочие ресурсы в соответствующем порядке с учетом атрибутов заданий – имен пользователей, приоритетов заданий, сложности задач и вычислений и др.

Режим обработки данных порождает соответствующий *режим функционирования системы*, проявляющийся в порядке инициирования задач и представлении одним задачам преимущественного права на использование ресурсов, в организации ввода данных, хранения программ в оперативной памяти, вывода данных и т. д. Порядок распределения ресурсов между заданиями влияет на время пребывания задания в системе, производительность системы, стоимость решения задач и другие характеристики системы и процессов обработки задач. Выбор того или иного режима обработки данных обусловлен необходимостью обеспечения требуемых характеристик системы и процессов обработки. В свою очередь характеристики системы влияют на способы взаимодействия пользователей с системой, а, следовательно, на интенсивность взаимодействия, продолжительность взаимодействия и т. д. Таким образом, режим обработки данных связан с организацией процесса функционирования системы и отражается в первую очередь на характеристиках системы.

Рассмотрим основные режимы обработки данных и их влияние на характеристики СОД.

Мультипрограммная обработка. В общем случае процесс решения задачи сводится к последовательности этапов процессорной обработки, ввода и вывода данных и обращений к внешним запоминающим устройствам. При этом задача в каждый момент времени обрабатывается, как правило, одним устройством, а остальные не могут использоваться до завершения работы этого устройства, следовательно, могут распределяться для выполнения других задач. Режим обработки, при котором в системе одновременно обрабатывается

несколько задач, называется *мультипрограммной обработкой* или, кратко, *мультипрограммированием*. При этом процессы обработки, относящиеся к разным задачам, одновременно выполняются различными устройствами системы, способными функционировать параллельно. В этом случае говорят, что система обработки данных функционирует в *мультипрограммном режиме*⁴. Цель мультипрограммирования – увеличение производительности системы.

Число задач, находящихся в системе, называется *уровнем мультипрограммирования*. Уровень мультипрограммирования влияет на производительность и время ответа системы следующим образом. На рис. 1.13 штриховыми линиями показана зависимость производительности системы λ , и среднего времени ответа U от уровня мультипрограммирования M . При изучении этих зависимостей удобно кривые $\lambda(M)$ и $U(M)$ представлять ломаными линиями, состоящими из двух прямых – асимптот и характеризующими верхнюю и нижнюю оценку: $\lambda(M)$ и $U(M)$ соответственно. В однопрограммном режиме ($M=1$) время ответа $U=U_1=\mathcal{G}$, где \mathcal{G} определяется по формуле (1.7). При этом производительность $\lambda=\lambda_1=1/U_1$.

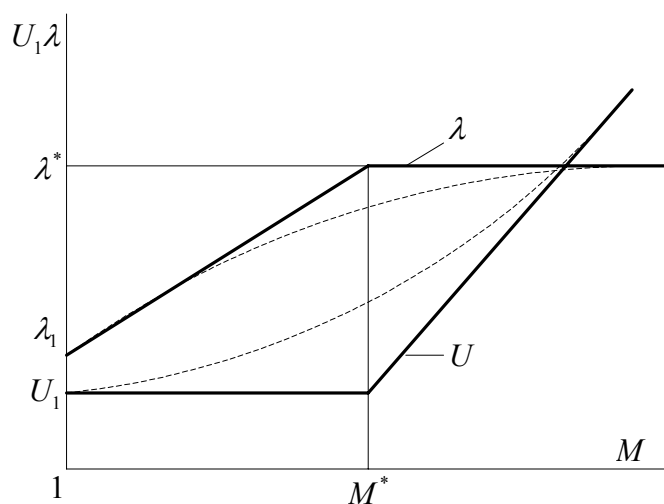


Рис. 1.13. Влияние уровня мультипрограммирования на производительность и время ответа

С увеличением уровня мультипрограммирования M увеличивается вероятность того, что большее число устройств одновременно занято выполнением задач. Но вместе с тем вероятность того, что несколько задач одновременно обращаются к одному устройству, достаточно мала, и поэтому время ожидания оказывается незначительным. Однако при уровне мультипрограммирования $M=M^*$ возникает ситуация, когда, по крайней мере, одно устройство оказывается полностью загруженным. Дальнейшее увеличение числа задач не приводит к росту производительности λ , которая определяется производительностью λ^* этого устройства, но при $M>M^*$ начинает резко возрастать время ответа U , поскольку все большее число задач ожидает момента освобождения устройств⁵. Значение M^* называется

⁴ В тех случаях, когда необходимо подчеркнуть, что параллелизм в работе системы достигается именно за счет одновременной обработки нескольких независимых задач, а не параллельных процессов решения одной задачи, вместо термина «мультипрограммный режим» используется термин «мультизадачный режим».

⁵ Полезно сопоставить характер изменения времени ответа на рис. 1.12 и 1.13. Ка рис. 1.12 при интенсивности входного потока задан $A \rightarrow X^*$ время от вета $1/\lambda^* \rightarrow \infty$, а на рис, 1.13 при уровне мультипрограммирования $M \rightarrow M^*$ время ответа $\{ \rightarrow \infty \}$, хотя в обоих случаях производительность системы $X \rightarrow \lambda^*$ ЯЛ Дело в том, что при $\lambda > X^*$ система функционирует в нестационарном режиме, когда в единицу времени поступает $(A - X^*)$ заданий, которые не могут быть обработаны, если $A > \lambda^*$. Поэтому теоретически число заданий в системе может оказаться сколь угодно большим и в пределе, при $\lambda \rightarrow \infty$, – бесконечным. На рис, 1.13 система всегда функционирует в стационарном режиме с постоянным числом задач M , находящихся в ней. Поэтому время ответа во всех случаях конечно.

точкой насыщения мультипрограммной смеси, а также точкой насыщения системы и зависит в первую очередь от числа устройств, которые в составе системы могут функционировать параллельно. Чем больше число устройств, тем больше M^* . Кроме того, на значение M^* существенно влияют свойства задач. Если задачи преимущественно используют одно устройство, то значение M^* невелико и может быть равным единице. Если задачи загружают все устройства, то значение M^* определяется числом устройств в системе.

Работа системы при уровне мультипрограммирования $M > M^*$ неэффективна, поскольку нет выигрыша в производительности и увеличивается время ответа.

Производительность λ и среднее время ответа U связаны между собой зависимостью

$$\lambda = M / U \quad (1.9)$$

которая называется *формулой Литтла* и является фундаментальным законом теории массового обслуживания.

В системе, состоящей из N устройств, загрузка которых равна ρ_1, \dots, ρ_N , среднее число задач m , выполняемых одновременно в мультипрограммном режиме, равно суммарной загрузке устройств:

$$m = \sum_{i=1}^N \rho_i \quad (1.10)$$

Остальные $M - m$ задач находятся в состоянии ожидания. Число одновременно выполняемых задач, определяемое (1.10), называется *коэффициентом мультипрограммирования* и равно отношению производительности системы в мультипрограммном режиме λ к производительности в однопрограммном режиме $\lambda_1 = 1/\Theta$, если затраты ресурсов на организацию мультипрограммирования возрастают пропорционально числу одновременно выполняемых задач, т. е.

$$m = \lambda / \lambda_1 \quad (1.11)$$

Таким образом, коэффициент мультипрограммирования m является показателем увеличения производительности системы за счет мультипрограммирования. Из (1.10) следует, что коэффициент мультипрограммирования

$$1 \leq m \leq N \quad (1.12)$$

где N – число устройств системы, способных функционировать параллельно с каждым из $N-1$ остальных устройств. При этом предполагается, что система работает без отказов.

Оперативная и пакетная обработка данных. Применительно к СОД, предназначенным для информационного обслуживания пользователей (но не технических объектов и систем), принято выделять два режима обработки данных: оперативную и пакетную обработку. *Оперативная обработка данных* характеризуется: 1) малым объемом вводимых – выводимых данных и вычислений, приходящимся на одно взаимодействие пользователя с системой (на одну задачу); 2) высокой интенсивностью взаимодействия и вытекающим отсюда требованием уменьшения времени ответа. Оперативная обработка необходима в системах банковских, резервирования билетов, справочных и др. Пакетная обработка данных характеризуется: 1) большим объемом вводимых – выводимых данных и вычислений, приходящимся на одно взаимодействие пользователя с системой (на одну задачу); 2) низкой интенсивностью взаимодействия и допустимостью большого времени ответа. Пакетная обработка типична для вычислительных центров научно-технического профиля, систем обработки учетно-статистических данных, результатов геофизических измерений и т. д.

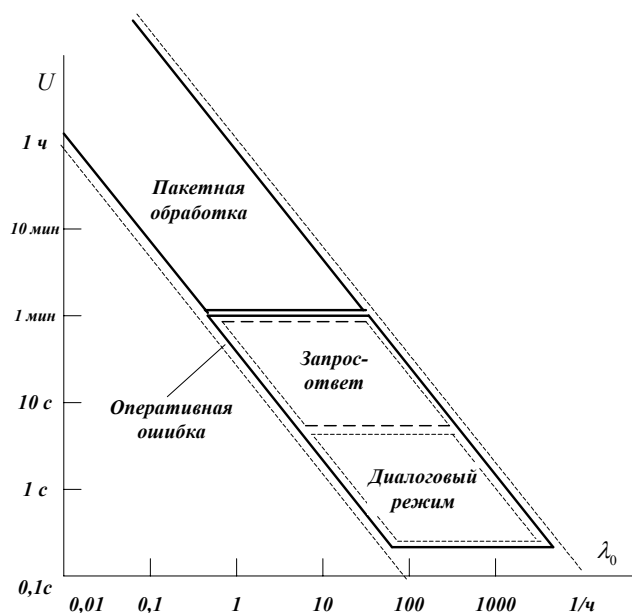


Рис. 1.14. Время ответа и интенсивность взаимодействия пользователей с системой для разных режимов обработки

Области типичных значений времени ответа U и интенсивности λ_0 взаимодействия пользователя с системой, соответствующие оперативной и пакетной обработке, представлены на рис. 1.14. Произведение $\rho_0 = \lambda_0 U$ определяет нагрузку, создаваемую пользователем. Значение ρ_0 можно рассматривать, во-первых, как долю времени, в течение которого пользователь обслуживается системой, и, во-вторых, как вероятность того, что в произвольный момент времени система инициирована пользователем, т. е. обслуживает его. Значение $1 - \rho_0$ определяет долю времени, в течение которого пользователь не взаимодействует с системой, и одновременно вероятность этого состояния. Так, для пакетной обработки ($\lambda_0 = 0,2ч^{-1}$ и $U = 0,15ч$) типичная нагрузка, создаваемая одним пользователем, $\rho_0 = 0,2 \cdot 0,15 = 0,03$.

В рамках оперативной обработки выделяют два режима: *запрос – ответ* и *диалоговый*. Режим *запрос – ответ* характеризуется меньшей интенсивностью и большей продолжительностью взаимодействия по сравнению с диалоговым режимом. Типичный пример использования режима *запрос–ответ* – справочная служба на основе ЭВМ. При этом пользователь формирует текст запроса, который вводится в ЭВМ, и ответ должен быть получен за несколько десятков секунд. Работа в диалоговом режиме предполагает практически мгновенный контакт пользователя с системой, при котором система реагирует на действия пользователя с задержкой в несколько секунд или доли секунды. Наиболее жесткие ограничения возникают, когда система должна обслуживать элементарные манипуляции пользователя, работающего за терминалом: например, реагировать на нажатие каждой клавиши. В этом случае время ответа не должно превышать 0,1 с. Менее жестким является режим, когда система должна реагировать только на моменты окончания набора строк, обеспечивая время ответа, равное нескольким секундам. Быстрота реакции системы на действия пользователя является неременным условием диалогового режима.

Диалоговый режим создает максимальные удобства для пользователя, обеспечивая постоянный контроль вводимых данных (программ и исходных данных), минимальное время ответа, возможность оперативного вмешательства пользователя в процесс решения задачи и оперативный доступ пользователя к системе. За счет этого минимизируются потери из-за простоя пользователей в ожидании результатов, некорректных действий пользователей; или «неожиданного» поведения программ. Однако диалоговый режим обеспечивается за счет использования системы с большой производительностью, что требует больших капитальных

вложений. Кроме того, стоимость выполнения программы в диалоговом режиме больше, чем в пакетном, из-за немалых издержек, связанных с управлением процессами со стороны операционной системы. При выполнении расчетов, особенно сложных, по апробированным программам и методикам постоянный контакт пользователя с системой необходим только на этапе ввода данных. При их обработке и выводе результатов оперативный контакт с пользователем не нужен. В таких случаях наиболее экономичным способом обработки данных является пакетный режим. В пакетном режиме организация процесса в системе имеет целью не минимизацию времени ответа, а снижение стоимости обработки данных за счет эффективного использования ресурсов системы. В пакетном режиме управление процессами – выбор заданий из очереди на обработку и порядок выполнения задач – направлено на повышение производительности системы за счет формирования смеси задач, обеспечивающей максимальную загрузку по возможности всех ресурсов системы. В этом случае время ответа становится весьма значительным: десятки минут и часы.

Вычислительные системы и комплексы в подавляющем большинстве случаев (в сети всегда) используются в *режиме коллективного доступа*, при котором с системой взаимодействует коллектив пользователей. Выполнение заданий производится в мультипрограммном режиме. Обеспечение режимов пакетного, запрос–ответ и диалогового производится за счет соответствующих способов управления ресурсами и процессами, реализуемых управляющими программами операционной системы.

Обработка в реальном масштабе времени. В системах управления реальными объектами, построенных на основе ЭВМ, процесс управления сводится к решению фиксированного набора задач $A = \{A_1, \dots, A_M\}$. Каждая задача инициируется либо периодически, либо при возникновении определенных ситуаций в системе. При этом темп инициирования задач и время получения результатов вычислений жестко регламентируются динамическими свойствами управляемого объекта: технологической установки, подвижного объекта и др. Это означает, что на время решения задач управления налагаются ограничения $u_1 \leq U_1^*, \dots, u_M \leq U_M^*$, определяющие предельное допустимое время ответа U_1^*, \dots, U_M^* для задач A_1, \dots, A_M соответственно. Режим, при котором организация обработки данных подчиняется темпу процессов вне СОД, называется обработкой в *реальном масштабе времени* (РМВ).

Обработка в РАШ обеспечивается за счет: 1) выбора структуры СОД и быстродействия устройств в соответствии с задачами обработки A и требованиями к времени обработки U_1^*, \dots, U_M^* ; 2) способов организации процессов обработки, обеспечивающих требуемое время ответа $u_1 \leq U_1^*, \dots, u_M \leq U_M^*$ при ограниченной производительности устройств и заданной структуре СОД.

Режим телеобработки данных. *Телеобработка* (удаленная обработка) – режим обработки данных при взаимодействии пользователей с СОД через линии связи. Телеобработка рассматривается в качестве самостоятельного режима обработки данных по следующим причинам. Во-первых, удаленность пользователей от СОД и наличие между ними специфического средства передачи данных – линии связи – порождает необходимость в специальных действиях пользователей при организации доступа к системе и завершении сеанса работы. Во-вторых, наличие линий связи налагает ограничения на форму и время обмена данными между пользователями и СОД. Эти ограничения приводят к необходимости специальных способов организации данных и доступа к ним, что в свою очередь отражается на структуре прикладных программ, используемых в режиме телеобработки.

Режим телеобработки характеризуется, прежде всего, спецификой доступа пользователя к системе и системы к данным, передаваемым через удаленные терминалы, т. е. связан в первую очередь с организацией обработки данных внутри СОД. При этом пользователи могут работать с режимах пакетном, диалоговом или «запрос–ответ». Каждый из этих режимов характеризуется специфичным способом взаимодействия пользователей с системой и соответствующим временем ответа.

ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ

2.1. ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ИНФОРМАЦИИ

В первой главе было показано, что один из основных факторов, определяющих развитие вычислительной техники в целом и вычислительных систем в частности, – это высокая производительность. Общий метод увеличения производительности – организация параллельной обработки информации, т. е. одновременное решение задач или совмещение во времени этапов решения одной задачи.

Способы организации. Во всем многообразии способов организации параллельной обработки можно выделить три основных направления:

- 1) совмещение во времени различных этапов разных задач;
- 2) одновременное решение различных задач или частей одной задачи;
- 3) конвейерная обработка информации.

Первый путь – совмещение во времени этапов решения разных задач – это мультипрограммная обработка информации. Мультипрограммная обработка возможна даже в однопроцессорной ЭВМ и широко используется в современных СОД. Второй путь – одновременное решение различных задач или частей одной задачи – возможен только при наличии нескольких обрабатывающих устройств. При этом используются те или иные особенности задач или потоков задач, что позволяет осуществить тот или иной параллелизм. Можно выделить несколько типов параллелизма, отражающих эти особенности.

Естественный параллелизм независимых задач заключается в том, что в систему поступает непрерывный поток не связанных между собой задач, т. е. решение любой задачи не зависит от результатов решения других задач. В этом случае использование нескольких обрабатывающих устройств при любом способе комплексования (косвенном или прямом) повышает производительность системы.

Параллелизм независимых ветвей – один из наиболее распространенных типов параллелизма в обработке информации. Суть его заключается в том, что при решении большой задачи могут быть выделены отдельные независимые части – ветви программы, которые при наличии нескольких обрабатывающих устройств могут выполняться параллельно и независимо друг от друга. Двумя независимыми ветвями программы будем считать такие части задачи, при выполнении которых выполняются следующие условия:

ни одна из входных для ветви программы величин не является выходной величиной другой программы (отсутствие функциональных связей);

для обеих ветвей программы не должна производиться запись в одни и те же ячейки памяти (отсутствие связи по использованию одних и тех же полей оперативной памяти);

условия выполнения одной ветви не зависят от результатов или признаков, полученных при выполнении другой ветви (независимость по управлению);

обе ветви должны выполняться по разным блокам программы (программная независимость).

Хорошее представление о параллелизме независимых ветвей дает ярусно-параллельная форма программы, пример которой приведен на рис. 2.1. Программа представлена в виде совокупности ветвей, расположенных в нескольких уровнях – ярусах. Кружками с цифрами внутри обозначены ветви. Длина ветви представляется цифрой, стоящей около кружка. Стрелками показаны входные данные и результаты обработки. Входные данные обозначаются символом x , выходные данные – символом y . Символы x

имеют нижние цифровые индексы, означающие номера входных величин; символы y имеют цифровые индексы и внизу и вверху; цифра вверху соответствует номеру ветви, при выполнении которой получен данный результат, а цифра внизу означает порядковый номер результата, полученного при реализации данной ветви программы.

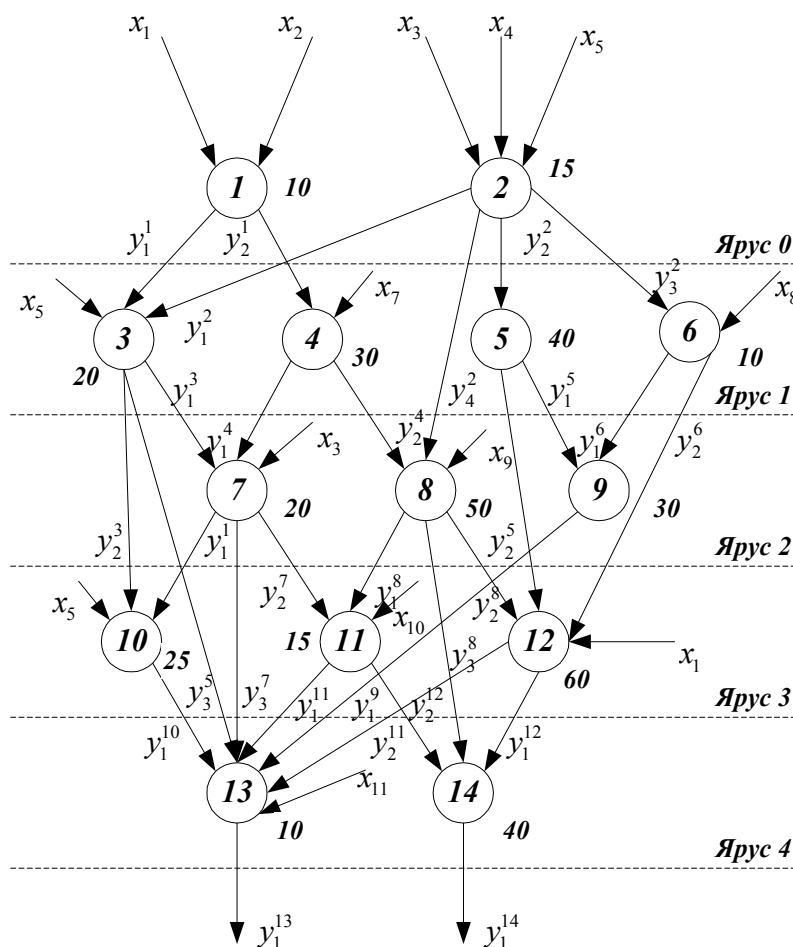


Рис. 2.1. Ярусно-параллельная форма программы

Изображенная на рисунке программа содержит 14 ветвей, расположенных на 5 ярусах. Ветви каждого яруса не связаны друг с другом, т. е. результаты решения какой-либо ветви данного яруса не являются входными данными для другой ветви этого же яруса. На этом же графе могут быть изображены и связи по управлению или памяти. В этом случае граф позволяет наглядно показать полностью независимые ветви. Для простоты изображения мы этого не делаем.

На примере этой, в общем достаточно простой программы, можно выявить преимущества вычислительной системы, включающей несколько обрабатывающих устройств, и проблемы, которые при этом возникают.

Примем, что длина i -й ветви представляется числом временных единиц t_i , которые требуются для ее исполнения. Тогда нетрудно подсчитать, что для исполнения всей программы потребуется $T = \sum_{i=1}^{N=14} t_i = 435$ единиц времени. Если представить, что программа

выполняется двумя обрабатывающими устройствами, работающими независимо друг от друга, то время решения задачи сократится. Однако это время, как нетрудно видеть, будет различным в зависимости от последовательности выполнения независимых ветвей.

Рассмотрим три варианта выполнения программы ярусно-параллельной формы (рис. 2.1).

Вариант 1. Процессор 1 выполняет ветви 1-4-5-9-10-13, процессор 2 – ветви 2-6-3-7-8-11-12-14. При этом процессор 1 затрачивает 260 единиц времени, из которых 55 простаивает, так как не готовы данные для ветви 13. Процессор 2 затрачивает 230 единиц времени.

Вариант 2. Процессор 1 выполняет ветви 1-4-5-9-10-11-13, процессор 2 - ветви 2-6-3-7-8-12-14. При этом процессор 1 затрачивает 245 единиц времени, из которых 25 простаивает по той же причине, что и в варианте 1. Процессор 2 затрачивает 215 единиц времени.

Вариант 3. Процессор 1 выполняет ветви 1-4-8-12-11-13, процессор 2 – ветви 2-5-6-3-7-9-10-14. При этом процессор 1 затрачивает 235, а процессор 2 – 205 единиц времени, из которых 5 он простаивает.

Сравнение этих вариантов показывает следующее. Во всех случаях время, через которое двухпроцессорная система выдает результаты y_1^{13} , y_1^{14} , существенно сокращается: вместо 435 единиц времени результаты выдаются в первом варианте через 260, во втором – через 245 и в третьем – через 235 единиц времени, т. е. в последнем случае время решения задачи уменьшается в 1,85 раза. Выигрыш по времени может существенно колебаться в зависимости от последовательности выполнения ветвей каждым процессором (читатель при желании может построить и другие варианты решения задачи), поэтому процессор должен выбирать новую ветвь с учетом этого обстоятельства. При решении задачи каждый процессор перед началом выполнения очередной ветви должен иметь информацию о готовности данных для этого.

Таким образом, для того чтобы с помощью нескольких обрабатывающих устройств решить задачу, имеющую независимые параллельные ветви, необходима соответствующая организация процесса, которая определяет пути решения задачи и вырабатывает необходимую информацию о готовности каждой ветви. Заметим, что все это относительно легко реализовать тогда, когда известна достаточно точно длительность выполнения каждой ветви. На практике это бывает крайне редко: в лучшем случае известна приближенная длина ветвей. Поэтому организация оптимального или близкого к оптимальному графика работы является достаточно сложной задачей.

Заметим, что, как правило, трудно избежать некоторых простоев, которые возникают из-за отсутствия исходных данных для выполнения той или иной ветви. Все это приводит в конечном счете к тому, что выигрыш в производительности системы несколько снижается. Следует отметить также и определенные сложности, связанные с выделением независимых ветвей при разработке программ. Вместе с тем при решении многих сложных задач только программирование с выделением независимых ветвей позволяет существенно сократить время решения. В частности, хорошо поддаются параллельной обработке такого типа задачи матричной алгебры, линейного программирования, спектральной обработки сигналов, прямые и обратные преобразования Фурье и др.

Параллелизм объектов или данных имеет место тогда, когда по одной и той же (или почти по одной и той же) программе должна обрабатываться некоторая совокупность данных, поступающих в систему одновременно.

Это могут быть, например, задачи обработки сигналов от радиолокационной станции: все сигналы обрабатываются по одной и той же программе. Другой пример – обработка информации от датчиков, измеряющих одновременно один и тот же параметр и установленных на нескольких однотипных объектах. Программы обработки данных могут быть различного объема и сложности, начиная от очень простых, содержащих несколько операций, до больших программ в сотни и тысячи операций.

Это могут быть и чисто математические задачи, например задачи векторной алгебры – операции над векторами и матрицами, характеризующиеся некоторой совокупностью чисел. Решение задачи при этом в значительной степени сводится к выполнению одинаковых операций над парами чисел двух аналогичных объектов. Так, например, сложение двух матриц размерностью $m \times n$ заключается в сложении соответствующих элементов этих матриц: $A + B = [a_{ik}] + [b_{ik}] = [a_{ik} + b_{ik}]$. При этом операция сложения должна быть проведена над $m \times n$ парами чисел. Произведение матрицы размерностью $m \times n$ на скаляр сводится к

выполнению $m \times n$ умножений элементов матрицы на скаляр: $\alpha A = \alpha [a_{ik}] = [\alpha a_{ik}]$ и т. д. Очевидно, все эти операции могут выполняться параллельно и независимо друг от друга несколькими обрабатывающими устройствами.

Третий путь параллельной обработки информации – *конвейерная обработка* – может быть реализован в системе и с одним процессором, разделенным на некоторое число последовательно включенных операционных блоков, каждый из которых специализирован на выполнении строго определенной части операции. При этом процессор работает таким образом: когда i -й операционный блок выполняет i -ю часть j -й операции, $(i-1)$ -й операционный блок выполняет $(i-1)$ -ю часть $(j+1)$ -й операции, а $(i+1)$ -й операционный блок выполняет $(i+1)$ -ю часть $(j-1)$ -й операции. В результате образуется своего рода конвейер обработки, который хорошо может быть проиллюстрирован следующим простым примером.



Этап	1	2	3	4	5	6	...	i	...	n	$n+1$	$n+2$	$n+3$
СП	$a_1 b_1$	$a_2 b_2$	$a_3 b_3$	$a_4 b_4$	$a_5 b_5$	$a_6 b_6$...	$a_i b_i$...	$a_n b_n$			
ВП		$a_1 b_1$	$a_2 b_2$	$a_3 b_3$	$a_4 b_4$	$a_5 b_5$...	$a_{i-1} b_{i-1}$...	$a_{n-1} b_{n-1}$	$a_n b_n$		
СМ			$a_1 b_1$	$a_2 b_2$	$a_3 b_3$	$a_4 b_4$...	$a_{i-2} b_{i-2}$...	$a_{n-2} b_{n-2}$	$a_{n-1} b_{n-1}$	$a_n b_n$	
НР				c_1	c_2	c_3	...	c_{i-3}	...	c_{n-3}	c_{n-2}	c_{n-1}	c_n

Рис. 2.2. Структурная схема (а) и временная диаграмма (б) конвейера операций

Операцию сложения двух чисел с плавающей запятой $A \cdot 2^x + B \cdot 2^y = C \cdot 2^{x/y}$ можно разделить на четыре последовательно исполняемых этапа, или шага: сравнение порядков (СП); выравнивание порядков – сдвиг мантииссы с меньшим порядком для выравнивания с мантииссой с большим порядком (ВП); сложение мантиисс (СМ); нормализация результата (НР). В соответствии с этим в составе процессора предусмотрены четыре операционных блока, соединенных последовательно и реализующих четыре вышеперечисленных шага операции сложения: блоки СП, ВП, СМ и НР (рис. 2.2, а). Примем, что время выполнения каждого шага равно соответственно 60, 100, 140, 100 нс. Таким образом, операция сложения будет выполняться последовательностью операционных блоков за время 400 нс.

Далее предположим, что возникает задача сложения двух векторов **А** и **В**, содержащих по i элементов с плавающей запятой. Очевидно, для решения этой задачи потребуется сложить два числа:

$$A + B = [a_i \cdot 2^x] + [b_i \cdot 2^y] = [c_i \cdot 2^{x/y}]$$

Выполним эти операции на процессоре, организовав обработку данных следующим образом. После того как блок СП выполнит свою часть операции над первой парой операндов, он передает результат в следующий блок – ВП, а в блок СП будет загружена очередная пара операндов. На следующем шаге блок ВП передает результат выполнения своей части операции в блок СМ и начнет обрабатывать вторую пару операндов и т. д. Для того чтобы не создавались очереди операндов на обработку, примем, что время выполнения каждого из этих этапов одинаково и равно максимальному значению $\tau = 140$ нс. В результате получим конвейер из четырех операционных блоков. Первый результат на выходе конвейера будет получен через $140 \times 4 = 560$ нс, т. е. несколько позже, чем если бы время выполнения всех этапов не выравнивалось. Однако все последующие результаты будут выдаваться через каждые 140 нс.

На рис. 2.2, б представлена временная диаграмма процесса. Общее время сложения двух векторов с помощью описанного конвейера $T_k = (n + m - 1)\tau$, где m – число операционных блоков. Если бы конвейер не использовался, то это время было бы равно $T_0 = n \sum_{i=1}^m \tau_i$ где τ_i – время выполнения i -го этапа обработки. Если применить конвейерную обработку к векторам, состоящим из 25 элементов, то получим $T_k = (25 + 4 - 1) \cdot 140 = 3920$ нс, $T_0 = 25 \cdot 400 = 10\,000$ нс.

Нетрудно заметить, что чем длиннее цепочка данных и чем на большее число этапов (а следовательно, и операционных блоков) разбивается операция (при той же длительности ее выполнения), тем больший эффект от использования конвейера может быть получен.

В приведенном выше примере было рассмотрено конвейерное выполнение арифметических операций, но идея конвейера может быть распространена и на выполнение команд. Надо сказать, что конвейер команд применяется уже давно в обычных ЭВМ. При этом цикл выполнения команды разбивается на ряд этапов, например формирование адреса команды (ФАК), выборка команды из памяти (ВК), расшифровка кода операции (РКО), формирование адреса операнда (ФАО), выборка операнда из памяти (ВО) и, наконец, арифметическая или логическая операция (АЛО). В устройстве управления предусматриваются блоки, которые независимо друг от друга и параллельно могут выполнять указанные этапы. Временная диаграмма показана на рис. 2.3. Для простоты время выполнения каждого этапа принято одинаковым (что, вообще говоря, не обязательно и не всегда делается).

Этап	1	2	3	4	5	6	7	8
ФАК	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8
ВК		K_1	K_2	K_3	K_4	K_5	K_6	K_7
РКО			K_1	K_2	K_3	K_4	K_5	K_6
ФАО				K_1	K_2	K_3	K_4	K_5
ВО					K_1	K_2	K_3	K_4
АЛО						K_1	K_2	K_3

Рис. 2.3. Временная диаграмма конвейера команд

Таким образом, если в конвейере арифметических операций происходит параллельная обработка m пар операндов, то в конвейере команд происходит совмещение во время выполнения l операций (l – число этапов, на которое разбито выполнение команды), что позволяет существенно увеличить производительность такой конвейерной системы.

К сожалению, выигрыш по производительности в l раз практически невозможен, так как может быть получен только при выполнении программы без условных переходов. Наличие условных переходов сразу нарушает работу конвейера и приводит к «холостым» пробегам конвейера, когда по выработанному в команде K_i признаку результата надо перейти к выполнению не K_{i+1} -й команды, а совершенно другой, что вызывает необходимость очистки всех блоков и загрузки их другой операцией. В реальных ЭВМ и системах применяются различные приемы, позволяющие определять признак перехода возможно раньше, однако совсем исключить влияние условных переходов не удастся. Тем не менее для определенных задач, где имеют место цепочки команд без таких переходов, выигрыш в производительности конвейерного процессора команд получается значительным. Как и в конвейере арифметических операций, выигрыш в производительности получается тем больше, чем длиннее участки программы без условных переходов и чем больше предусматривается независимость этапов (и, следовательно, блоков устройства управления) при выполнении команды.

Разумеется, в вычислительных системах можно одновременно использовать и конвейер команд, и конвейер арифметических операций, и даже несколько параллельно работающих конвейеров команд и арифметических операций. В последнем случае может быть получена очень высокая производительность системы. Именно по такому принципу построены самые быстродействующие вычислительные системы, которые будут рассмотрены несколько позже.

Классификация систем параллельной обработки. Процесс решения задачи можно представить как воздействие определенной последовательности команд программы (потока команд) на соответствующую последовательность данных (поток данных), вызываемых этой последовательностью команд. Различные способы организации параллельной обработки информации можно представить как способы организации одновременного воздействия одного или нескольких потоков команд на одни или несколько потоков данных.

Для такой классификации оказывается полезным ввести понятие множественности потоков команд и данных. Под *множественным потоком команд или данных* будем понимать наличие в системе нескольких последовательностей команд, находящихся в стадии реализации, или нескольких последовательностей данных, подвергающихся обработке командами.

Исходя из возможности существования одиночных и множественных потоков, все системы могут быть разбиты на четыре больших класса.

1. Системы с одиночным потоком команд и одиночным потоком данных (ОКОД).
2. Системы с множественным потоком команд и одиночным потоком данных (МКОД).
3. Системы с одиночным потоком команд и множественным потоком данных (ОКМД).
4. Системы с множественным потоком команд и множественным потоком данных (МКМД).

Реализация описанных способов организации параллельной обработки информации хорошо укладывается в представленную классификацию, и каждому классу систем присущи вполне определенные способы.

Системы класса ОКОД. Системы этого класса – обычные однопроцессорные ЭВМ (рис. 2.4, *a*), включающие в себя запоминающее устройство (ЗУ) для команд и данных (оно чаще всего бывает общим) и один процессор, содержащий арифметическо-логическое устройство (АЛУ) и устройство управления (УУ). В современных системах этого класса наиболее широко используется первый путь организации параллельной обработки – совмещение во времени различных этапов решения разных задач, при котором в системе одновременно работают различные устройства: ввода, вывода и собственно обработки информации. В общем виде это достаточно подробно рассмотрено в предыдущей главе, однако практическая реализация этого принципа предусматривает и более широкое совмещение работы различных устройств. В отношении ввода–вывода информации – это введение нескольких одновременно работающих каналов ввода–вывода, а также нескольких устройств одного типа: перфокарточных устройств ввода и вывода, печатающих устройств, различного рода накопителей и др. Введение большого числа параллельно работающих периферийных устройств позволяет существенно сократить время на ввод информации и оперативное запоминающее устройство: (ОЗУ), уменьшая общее время решения задачи, и до некоторой степени сгладить разрыв между скоростями работы центральных (процессор и ОЗУ) и периферийных устройств. Значительный эффект в производительности однопроцессорной системы дает разделение ОЗУ на несколько модулей, функционально самостоятельных, что позволяет им работать независимо друг от друга. Производительность увеличивается за счет уменьшения простоев устройств из-за так называемых конфликтов при обращении к ОЗУ.

При параллельной работе многих устройств ввода–вывода и процессора неизбежны ситуации, когда нескольким устройствам требуется обращение к ОЗУ для записи или чтения

информации. Такая ситуация и называется *конфликтной*. Естественно, разрешить этот конфликт можно только путем введения системы приоритетов, которая устанавливает определенную очередность удовлетворения запросов в ОЗУ. При этом неизбежны очереди, а, следовательно, и простои устройств, не имеющих высшего приоритета. В наибольшей степени от этого страдает процессор, так как по принципу своей работы он может ожидать в очереди обслуживания сколь угодно долго, в то время как большинство периферийных устройств, в основном электромеханических, ждать долго не могут или это будет приводить к слишком большим потерям времени. Так, например, если вовремя не считать информацию с накопителя на магнитном диске, то придется потерять несколько миллисекунд (один оборот диска) до следующего момента, когда эта информация может быть считана. При наличии нескольких модулей ОЗУ с независимым управлением есть определенная вероятность того, что различные устройства будут обращаться к различным модулям, а следовательно, очередь к ОЗУ разделится на несколько меньших очередей и время ожидания (т. е. простой устройства) в очереди будет уменьшено. Кроме совмещения во времени различных этапов обработки информации в системах класса ОКОД существенное увеличение производительности достигается за счет введения конвейерной обработки, точнее конвейера команд.

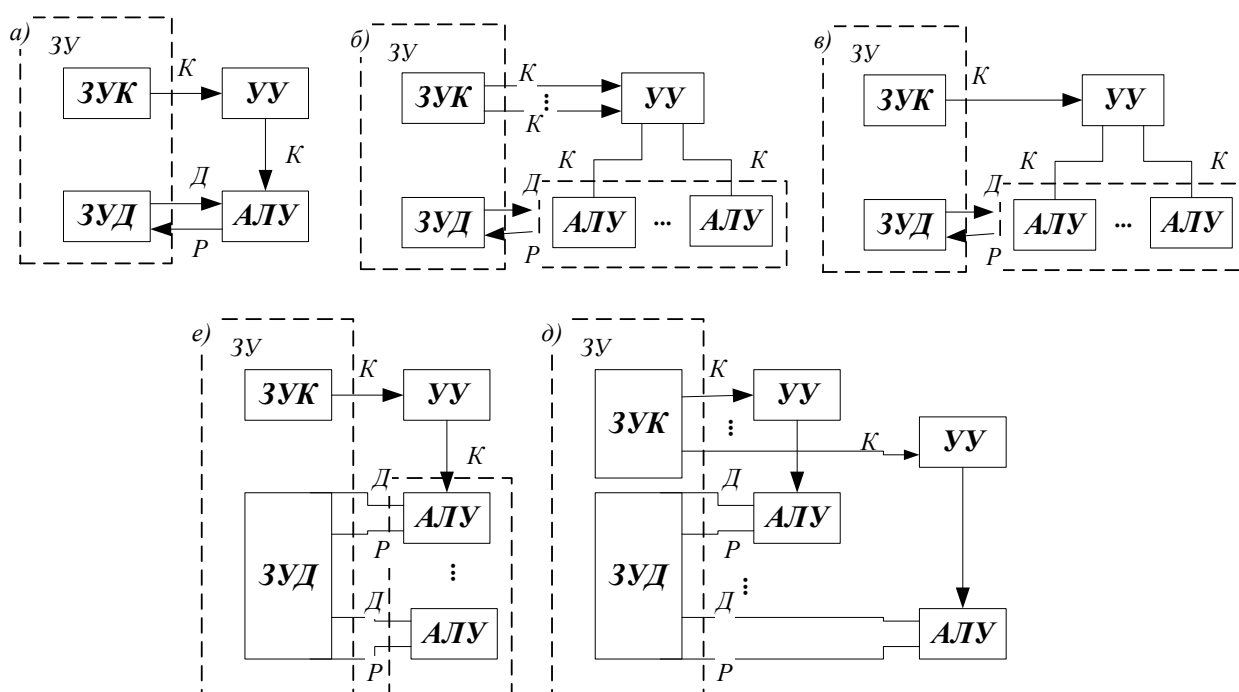


Рис. 2.4. Система ОКОД (а), МКОД (б, в), ОКМД (г), МКМД (д)
 Д – данные; К – команды; Р - результат

По существу конвейер команд – это также совмещение во времени работы нескольких различных блоков, выполняющих отдельные части общей операции. Если совмещение во времени работы различных устройств можно назвать «макросовмещением», то конвейер является как бы «микросовмещением». При организации конвейера команд очень часто применяются также способы увеличения производительности, базирующиеся на использовании многомодульных ОЗУ. Во-первых, если программы и данные размещать в разных модулях памяти, то это позволит совмещать во времени выборку команды и операнда, что при выполнении ОЗУ в виде одного функционального устройства было бы невозможно. Во-вторых, используется тот факт, что команды и данные обычно при обработке выбираются из некоторой последовательности ячеек памяти с последовательно возрастающими адресами. Если организовать ОЗУ таким образом, что все четные адреса будут принадлежать одному модулю ОЗУ, а все нечетные – другому, и сдвинуть начало

цикла работы этих двух модулей на $\frac{1}{2}$ цикла, то при выполнении программы среднее время обращения к ОЗУ существенно уменьшается (в пределе в 2 раза). Этот принцип может быть распространен и на большее число модулей ОЗУ с независимым управлением. При N модулях ОЗУ среднее время обращения к ОЗУ оказывается равным $1/N$ -й цикла ОЗУ. Такая память называется памятью с чередованием адресов или расслоением обращений.

В системах класса ОКОД возможна реализация и еще одного способа увеличения производительности – конвейер арифметических и логических операций, который вполне вписывается в этот класс систем, так как поток команд остается один: просто команды разбиваются на некоторое число микроопераций, образуя, таким образом, несколько потоков микрокоманд. По этой причине системы с конвейером арифметических и логических операций относятся к следующему классу – МКОД.

Детальное рассмотрение систем класса ОКОД не входит в нашу задачу – это делается в курсе «Теория и проектирование ЭВМ», одним из разделов которого является подробное изучение всех перечисленных выше способов повышения производительности ЭВМ.

Системы класса МКОД. Структуру систем этого класса можно представить в виде схемы, изображенной на рис. 2.4, б: несколько потоков команд воздействуют на единственный поток данных. Однако не существует такого класса задач, в которых одна и та же последовательность данных подвергалась бы обработке по нескольким разным программам. По этой причине в чистом виде такая схема до сих пор не реализована.

Как уже отмечалось, реализована другая схема обработки, представленная на рис. 2.4, в. Здесь один поток команд K разделяется устройством управления на несколько потоков микрооперации, каждая из которых реализуется специализированным, настроенным на выполнение именно данной микрооперации, устройством. Поток данных проходит последовательно через, все (или часть) этих специализированных ЛЛУ. Именно такого класса системы принято называть *конвейерными* или системами с *магистральной обработкой* информации.

Разумеется, в системах МКОД с целью достижения высокой производительности используется не только конвейер операции. Обычно в таких системах используется и конвейер команд, и различные способы совмещения работы многих устройств. Однако при этом главным, определяющим признаком является наличие конвейера арифметических и логических операций. Заметим также, что системы этого класса развивают максимальную производительность только при решении задач определенного типа, в которых существуют длинные последовательности (цепочки) однотипных операций над достаточно большой последовательностью данных, т. е. когда имеет место параллелизм объектов или данных.

Системы класса ОКМД. Системы этого класса также ориентированы на использование параллелизма объектов или данных для повышения производительности. Обобщенная структура представлена на рис. 2.4, г. В этой системе по одной и той же (или почти по одной и той же) программе обрабатывается несколько потоков данных, каждый из этих потоков обрабатывается своим АЛУ, работающим однако под общим управлением, за счет чего и достигается высокая производительность системы.

Общая схема, представленная на рис. 2.4, г, может реализоваться разными способами. Так, например, АЛУ может представлять собой достаточно сложное устройство, содержащее обрабатывающий процессор и оперативное ЗУ. В этом случае поток данных в каждый процессор поступает из собственного ЗУ. Управление и память команд реализуются отдельной ЭВМ, управляющей ансамблем процессоров. Память данных может иметь не только адресную выборку, но и ассоциативную, т. е. по содержимому памяти.

В системах класса ОКМД могут использоваться для достижения высокой производительности и другие пути параллельной обработки, однако определяющим является одновременная обработка нескольких потоков несколькими процессорами. Поэтому все системы класса ОКМД можно разделить на матричные и ассоциативные.

Системы класса МКМД. Возможны два способа построения систем МКМД: в виде совокупности элементарных систем (рис. 2.4, а) и по схеме, приведенной на рис. 2.4, д. На

первый взгляд разница незначительная: в варианте *a* для каждой последовательности команд и данных имеется собственное ЗУ, в варианте *б* все команды и все данные размещаются в общих ЗУ. Это различие приводит однако к двум сильно отличающимся по построению и организации вычисления типам вычислительных систем и комплексов: многомашинным (рис. 2.4, *a*) и многопроцессорным (рис. 2.4, *б*). В многомашинном варианте вся система как бы распадается на несколько независимых систем класса ОКОД, т. е. по существу самостоятельных ЭВМ со всеми особенностями, присущими им. При этом обычно существуют определенные связи между ЭВМ, объединяющие эти ЭВМ в систему. В многопроцессорном варианте система (комплекс) достаточно жестко связана общей памятью команд и данных и, хотя процессоры системы имеют достаточную самостоятельность, в системе организуется совместная их работа.

Многомашинные системы (комплексы) в наилучшей степени приспособлены для решения потока независимых задач. При этом объединение в систему нескольких ЭВМ существенно увеличивает производительность системы. Многопроцессорные системы (комплексы) являются достаточно универсальными и обеспечивают повышение производительности при использовании всех видов параллелизма, но, пожалуй, наибольший эффект достигается при задачах, характеризующихся параллелизмом независимых ветвей.

Из всех рассмотренных систем наиболее универсальными в отношении класса решаемых задач являются системы МКМД. Их программное обеспечение не ориентируется, как правило, на решение только определенного класса задач. Таким образом, системы типа МКМД строятся как вычислительные комплексы. Многопроцессорные и многомашинные комплексы имеют наиболее широкое распространение, и по этой причине мы именно с них начинаем подробное рассмотрение.

Вычислительные системы класса МКОД и ОКМД, ориентированные на решение задач определенных классов, детально рассматриваются в следующей главе.

2.2. МНОГОМАШИННЫЕ КОМПЛЕКСЫ

Многомашинный вычислительный комплекс (ММВК) – комплекс, включающий в себя две или более ЭВМ (каждая из которых имеет процессор, ОЗУ, набор периферийных устройств и работает под управлением собственной операционной системы), связи между которыми обеспечивают выполнение функций, возложенных на комплекс.

Цели, которые ставятся при объединении ЭВМ в комплекс, могут быть различными, и они определяют характер связей между ЭВМ. Чаще всего основной целью создания ММВК является или увеличение производительности, или повышение надежности, или одновременно и то и другое. Однако при достижении одних и тех же целей связи между ЭВМ могут существенно различаться.

По характеру связей между ЭВМ комплексы можно разделить на три типа: косвенно-, или слабосвязанные; прямосвязанные; сателлитные.

В *косвенно-, или слабосвязанных, комплексах* ЭВМ связаны друг с другом только через внешние запоминающие устройства (ВЗУ). Для обеспечения таких связей используются устройства управления ВЗУ с двумя и более входами. Структурная схема такого ММВК приведена на рис. 2.5. Заметим, что здесь и далее для простоты приводятся схемы для двухмашинных комплексов. При трех и более ЭВМ комплексы строятся аналогичным образом. В косвенно-связанных комплексах связь между ЭВМ осуществляется только на информационном уровне. Обмен информацией осуществляется в основном по принципу «почтового ящика», т. е. каждая из ЭВМ помещает в общую внешнюю память информацию, руководствуясь собственной программой, и соответственно другая ЭВМ принимает эту информацию, исходя из своих потребностей. Такая организация связей обычно используется в тех случаях, когда ставится задача повысить надежность комплекса путем резервирования ЭВМ. В этом случае ЭВМ, являющаяся основной, решает заданные задачи, выдает результаты и постоянно оставляет в общем ВЗУ всю информацию, необходимую для продолжения решения с любого момента времени. Вторая ЭВМ,

являющаяся резервной, может находиться в состоянии ожидания, с тем чтобы в случае выхода из строя основной ЭВМ, по сигналу оператора начать выполнение функций, используя информацию, хранимую в общем ВЗУ основной ЭВМ.

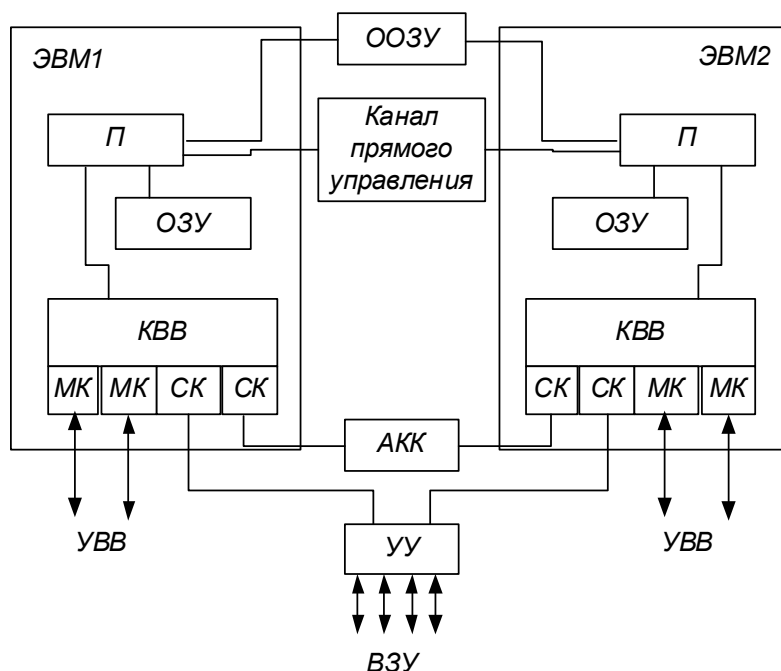


Рис. 2.5. Связи между ЭВМ и ММВК

При такой связи может быть несколько способов организации работы комплекса.

1. Резервная ЭВМ находится в выключенном состоянии (ненагруженный резерв) и включается только при отказе основной ЭВМ. Естественно, для того чтобы резервная ЭВМ начала выдавать результаты вместо основной, потребуется определенное время, которое определяется временем, необходимым для включения ЭВМ, входением ее в режим, а также временем, отводимым для проверки ее исправности. Это время может быть достаточно большим. Такая организация возможна, когда система, в которой работает ЭВМ, не критична по отношению к некоторым перерывам или остановкам в процессе решения задач. Это обычно имеет место в случаях, когда ЭВМ не выдает управляющую информацию.

2. Резервная ЭВМ находится в состоянии полной готовности и в любой момент может заменить основную ЭВМ (нагруженный резерв), причем либо не решает никаких задач, либо работает в режиме самоконтроля, решая контрольные задачи. В этом случае переход в работе от основной к резервной ЭВМ может осуществляться достаточно быстро, практически без перерыва в выдаче результатов. Однако следует заметить, что основная ЭВМ обновляет в общем ВЗУ информацию, необходимую для продолжения решения, не непрерывно, а с определенной дискретностью, поэтому резервная ЭВМ начинает решать задачи, возвращаясь на некоторое время назад. Такая организация допустима и в тех случаях, когда ЭВМ работает непосредственно в контуре управления, а управляемым процесс достаточно медленным и возврат во времени не оказывает заметного влияния.

При организации работы по первому и второму вариантам ЭВМ используются нерационально: одна ЭВМ всегда простаивает. Простоев можно избежать, загружая ЭВМ решением каких-то вспомогательных задач, не имеющих отношения к основному процессу. Это повышает эффективность системы – производительность практически удваивается.

3. Для того чтобы полностью исключить перерыв в выдаче результатов, обе ЭВМ, и основная и резервная, решают одновременно одни и те же задачи, но результаты выдает только основная ЭВМ, а в случае выхода ее из строя результаты начинает выдать резервная ЭВМ. При этом общее ВЗУ используется только для взаимного контроля. Иногда такой комплекс дополняется устройством для сравнения результатов с целью контроля. Если

при этом используются три ЭВМ, то возможно применение метода голосования, когда окончательный результат выдается только при совпадении результатов решения задачи не менее чем от двух ЭВМ. Это повышает и надежность комплекса в целом, и достоверность выдаваемых результатов. Разумеется, в этом варианте высокая надежность и оперативность достигается весьма высокой ценой – увеличением стоимости системы.

Следует обратить внимание, что при любой организации работы и слабосвязанном ММВК переключение ЭВМ осуществляется либо по командам оператора, либо с помощью дополнительных средств, осуществляющих контроль исправности ЭВМ и вырабатывающих необходимые сигналы. Кроме того, быстрый переход к работе с основной на резервную ЭВМ возможен лишь при низкой эффективности использования оборудования.

Существенно большей гибкостью обладают прямосвязанные ММВК. В прямосвязанных комплексах существуют три вида связей (рис. 2.5): общее ОЗУ (ООЗУ); прямое управление, иначе связь процессор (П) – процессор; адаптер канал – канал (АКК).

Связь через общее ОЗУ гораздо сильнее связи через ВЗУ. Хотя первая связь также носит характер информационной связи и обмен информацией осуществляется по принципу «почтового ящика», однако, вследствие того, что процессоры имеют прямой доступ к ОЗУ, все процессы в системе могут протекать с существенно большей скоростью, а разрывы в выдаче результатов при переходах с основной ЭВМ на резервную сокращаются до минимума. Недостаток связи через общее ОЗУ заключается в том, что при выходе из строя ОЗУ, которое является сложным электронным устройством, нарушается работа всей системы. Чтобы этого избежать, приходится строить общее ОЗУ из нескольких модулей и резервировать информацию. Это, в свою очередь, приводит к усложнению организации вычислительного процесса в комплексе и в конечном счете к усложнению операционных систем. Следует отметить также и то, что связи через общее ОЗУ существенно дороже, чем через ВЗУ.

Непосредственная связь между процессорами – канал прямого управления – может быть не только информационной, но и командной, т. е. по каналу прямого управления один процессор может непосредственно управлять действиями другого процессора. Это, естественно, улучшает динамику перехода от основной ЭВМ к резервной, позволяет осуществлять более полный взаимный контроль ЭВМ. Вместе с тем передача сколь угодно значительных объемов информации по каналу прямого управления нецелесообразна, так как в этом случае решение задач прекращается: процессоры ведут обмен информацией.

Связь через адаптер канал – канал в значительной степени устраняет недостатки связи через общее ОЗУ и вместе с тем почти не уменьшает возможностей по обмену информацией между ЭВМ по сравнению с общим ОЗУ. Сущность этого способа связи заключается в том, что связываются между собой каналы двух ЭВМ с помощью специального устройства – адаптера. Обычно это устройство подключается к селекторным каналам ЭВМ. Такое подключение адаптера обеспечивает достаточно быстрый обмен информацией между ЭВМ, при этом обмен может производиться большими массивами информации. В отношении скорости передачи информации связь через АКК мало уступает связи через общее ОЗУ, а в отношении объема передаваемой информации – связи через общее ВЗУ. Функции АКК достаточно просты: это устройство должно обеспечивать взаимную синхронизацию работы двух ЭВМ и буферизацию информации при ее передаче. Хотя функции АКК и его структура (рис. 2.5) достаточно просты, однако большое разнообразие режимов работы двух ЭВМ и необходимость реализации этих режимов существенно усложняет это устройство.

Прямосвязанные комплексы позволяют осуществлять все способы организации ММВК, характерные для слабосвязанных комплексов. Однако за счет некоторого усложнения связей эффективность комплексов может быть значительно повышена. В частности, в прямосвязанных комплексах возможен быстрый переход от основной ЭВМ к резервной и в тех случаях, когда резервная ЭВМ загружена собственными задачами. Это позволяет обеспечивать высокую надежность при высокой производительности.

В реальных комплексах одновременно используется не один вид связи между ЭВМ, а два или более. В том числе очень часто в прямых комплексах присутствует и косвенная связь через ВЗУ.

Для комплексов с *спутниковыми ЭВМ* характерным является не способ связи, а принципы взаимодействия ЭВМ. Структура связей в спутниковых комплексах не отличается от связей в обычных ММВК: чаще всего связь между ЭВМ осуществляется через АКК. Особенностью же этих комплексов является то, что в них, во-первых, ЭВМ существенно различаются по своим характеристикам, а во-вторых, имеет место определенная соподчиненность машин и различие функций, выполняемых каждой ЭВМ. Одна из ЭВМ, основная, является, как правило, высокопроизводительной и предназначается для основной обработки информации. Вторая, существенно меньшая по производительности, называется спутниковой или вспомогательной ЭВМ. Ее назначение – организация обмена информацией основной ЭВМ с периферийными устройствами, ВЗУ, удаленными абонентами, подключенными через аппаратуру передачи данных к основной ЭВМ. Кроме того, спутниковая ЭВМ может производить предварительную сортировку информации, преобразование ее в форму, удобную для обработки на основной ЭВМ, приведение выходной информации к виду, удобному для пользователя, и др. Спутниковая ЭВМ, таким образом, избавляет основную высокопроизводительную ЭВМ от выполнения многочисленных действий, которые не требуют ни большой разрядности, ни сложных операций, т. е. операций, для которых большая, мощная ЭВМ не нужна. Более того, с учетом характера выполняемых спутниковой машиной операций она может быть ориентирована на выполнение именно такого класса операций и обеспечивать даже большую производительность, чем основная ЭВМ.

Некоторые комплексы включают в себя не одну, а несколько спутниковых ЭВМ, при этом каждая из них ориентируется на выполнение определенных функций: например, одна осуществляет связь основной ЭВМ с устройствами ввода–вывода информации, другая – связь с удаленными абонентами, третья организует файловую систему и т. д.

Появление в последнее время дешевых и простых микро-ЭВМ в немалой степени способствует развитию спутниковых комплексов. Спутниковые комплексы решают только одну задачу: увеличивают производительность комплекса, не оказывая заметного влияния на показатели надежности.

Подключение спутниковых ЭВМ принципиально возможно не только через АКК, но и другими способами, однако связь через АКК наиболее удобна.

2.3. МНОГОПРОЦЕССОРНЫЕ КОМПЛЕКСЫ

Многопроцессорный вычислительный комплекс (МПВК) – это комплекс, включающий в себя два или более процессоров, имеющих общую оперативную память, общие периферийные устройства и работающих под управлением единой операционной системы (ОС), которая, в свою очередь, осуществляет общее управление техническими и программными средствами комплекса. Следует оговорить, что каждый из процессоров может иметь индивидуальные, доступные только ему ОЗУ и периферийные устройства. Все перечисленное весьма существенно, так как делает возможной гибкую организацию параллельной обработки информации и позволяет наиболее эффективно использовать все ресурсы комплекса.

На рис. 2.6 представлена упрощенная схема МПВК, содержащая три процессора, два модуля ОЗУ и одну подсистему ввода – вывода информации (ПВВ). Даже для такого простого варианта схема оказывается достаточно сложной, так как в МПВК должен быть обеспечен доступ любого процессора и любого канала ввода – вывода к любой ячейке ОЗУ, любого процессора к любому каналу и периферийному устройству. Если представить теперь, что процессоров существенно больше, что ОЗУ по соображениям надежности и удобства наращивания емкости выполнено в виде нескольких модулей, а подсистема ввода – вывода включает в себе несколько каналов и большое число периферийных устройств, то становится

ясным, насколько сложна топология МПВК. Если же учесть то обстоятельство, что для работы ОС аппаратные средства должны обеспечивать работу с переменными логическими адресами ОЗУ, периферийных устройств и каналов ввода – вывода, защиту памяти от взаимного влияния различных программ и возможность запуска одним процессором другого, сложность аппаратной реализации МПВК становится ясной в полной мере.

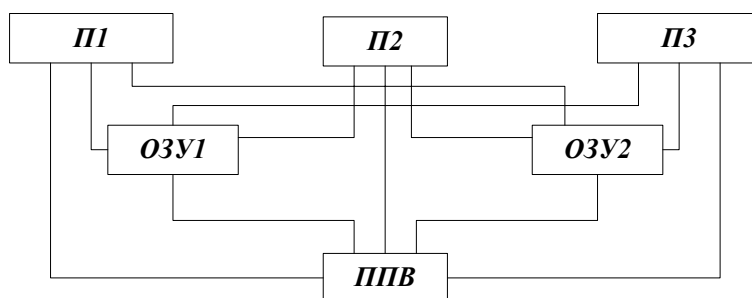


Рис. 2.6. Связи в МПВК

Непростые задачи возникают и при организации вычислительного процесса в МПВК., т. е. при построении ОС, которые являются основным средством организации всех процессов обработки информации в комплексе. Кроме обычных функций, выполняемых ОС при мультипрограммной обработке информации, возникают такие задачи, как распределение ресурсов и заданий между процессорами, синхронизация процессов при решении несколькими процессорами одной задачи, планирование с учетом оптимизации загрузки всех процессоров. При этом надо иметь в виду, что в процессе работы в комплексе возникает большое число конфликтных ситуаций, которые должны обрабатываться ОС. Эти и ряд других обстоятельств и факторов, связанных с обеспечением высокой надежности, делают ОС МПВК чрезвычайно сложной.

Однако, несмотря на все трудности, связанные с аппаратной и программной реализацией, МПВК получают все большее распространение, так как обладают рядом достоинств, основные из которых:

- высокая надежность и готовность за счет резервирования и возможности реконфигурации;

- высокая производительность за счет возможности гибкой организации параллельной обработки информации и более полной загрузки всего оборудования;

- высокая экономическая эффективность за счет повышения коэффициента использования оборудования комплекса.

Рассматривая процесс появления и развития МПВК, по-видимому, следует признать, что первоначально перед МПВК ставилась только задача обеспечения высокой надежности системы.

Неслучайно поэтому, что одним из первых (а может быть, и самым первым) МПВК был комплекс D-825, созданный фирмой «Барроуз» (США) в 1968 г. для систем военного назначения. Комплекс включал в себя 4 процессора, 16 модулей ОЗУ, 10 каналов ввода – вывода и до 256 периферийных устройств, т. е. был весьма представительным МПВК, по тем меркам. Надо сказать, что эта первая попытка была весьма удачной – поставленные задачи были полностью решены. Вместе с тем создание первых же МПВК выявили и возможности достижения с их помощью высокой производительности. В настоящее время МПВК чаще создаются именно с такой целью.

Типы структурной организации МПВК. Существует три типа структурной организации МПВК: с общей шиной; с перекрестной коммутацией; с многоходовыми ОЗУ.

В комплексах с *общей шиной* проблема связей всех устройств между собой решается крайне просто: все они соединяются общей шиной, выполненной в виде совокупности проводов или кабелей, по которым передаются информация, адреса и сигналы управления

(рис. 2.7, а). Интерфейс является односвязным, т. е. обмен информацией в любой момент времени может происходить только между двумя устройствами. Если потребность в обмене существует более чем у двух устройств, то возникает конфликтная ситуация, которая разрешается с помощью системы приоритетов и организации очередей в соответствии с этим. Обычно функции арбитра выполняет либо процессор, либо специальное устройство, которое регистрирует все обращения к общей шине и распределяет шину во времени между всеми устройствами комплекса.

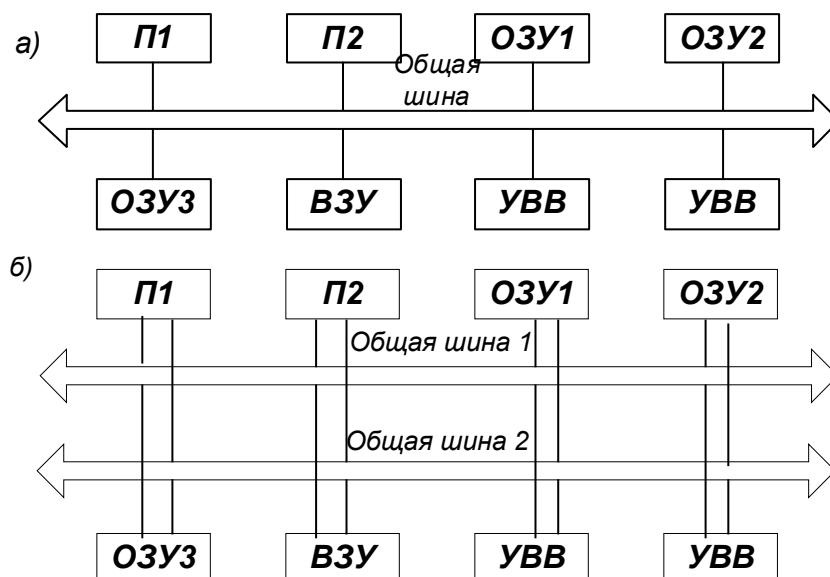


Рис. 2.7. МПВК с общей шиной

Несомненные достоинства структуры с общей шиной – простота, в том числе изменения комплекса, добавления или изъятия отдельных устройств, а также доступность модулей ОЗУ для всех остальных устройств. Следствием всего этого является достаточно низкая стоимость комплекса.

Вместе с тем комплексы с общей шиной не лишены определенных недостатков. Первый – невысокое быстродействие, так как одновременный обмен информацией возможен между двумя устройствами, не более. По этой причине в комплексах с общей шиной число процессоров не превосходит двух-четырех. Этот недостаток может быть несколько компенсирован путем использования общей шины с высоким быстродействием, большим, чем быстродействие входящих в комплекс устройств. Однако этот путь приводит к усложнению и удорожанию комплекса. Вторым недостатком МПВК с общей шиной заключается в относительно низкой надежности системы из-за наличия общего элемента – шины. Надо иметь в виду, что надежность общей шины определяется не только надежностью проводов и кабелей (их собственная надежность достаточно высока), но и надежностью всех соединений, входных и выходных цепей устройства. Отказ хотя бы одного из элементов приводит к отказу всего комплекса. Этот недостаток можно компенсировать за счет введения резервной шины (рис. 2.7, б). Хотя это несколько усложняет комплекс, однако надежность его существенно возрастает. Если же резервную шину сделать активной, т. е. работающей одновременно с основной, то можно не только повысить надежность, но и увеличить производительность комплекса за счет того, что обмен информацией может осуществляться одновременно между двумя парами устройств.

Общая шина может быть организована различными способами – принципиально так же, как и для однопроцессорных ЭВМ с общей шиной. Характерным примером является комплекс СМ-1420, в котором используется общая шина однопроцессорных ЭВМ этой системы.

Полностью лишены недостатков, присущих МПВК с общей шиной, МПВК с *перекрестной коммутацией*. Идея структурной организации таких ВК заключается в том, что все связи между устройствами осуществляются с помощью специального устройства – коммутационной матрицы (КМ) (рис. 2.8, а). Коммутационная матрица (КМ) позволяет связывать друг с другом любую пару устройств, причем таких пар может быть сколько угодно: связи не зависят друг от друга.

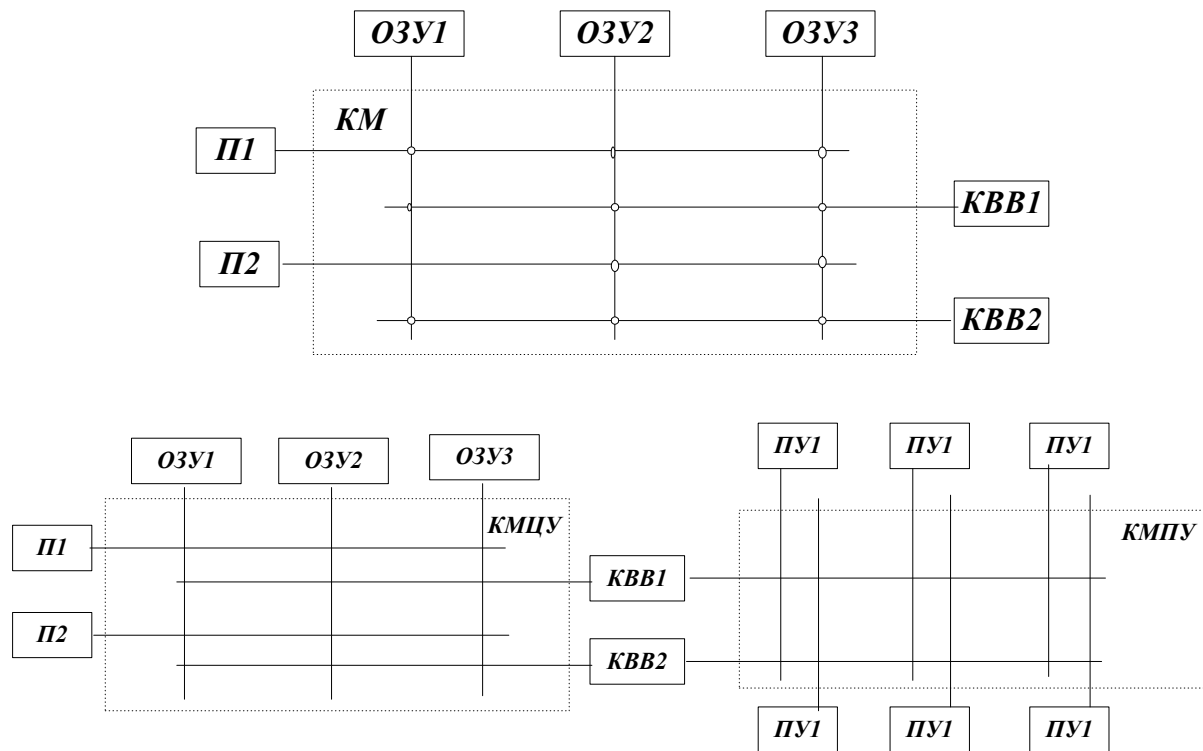


Рис. 2.8. МПВК с перекрестной коммутацией

В МПВК с перекрестной коммутацией нет конфликтов из-за связей, остаются только конфликты из-за ресурсов. Возможность одновременной связи нескольких пар устройств позволяет добиваться очень высокой производительности комплекса. Важно отметить и такое обстоятельство, как возможность установления связи между устройствами на любое, даже на длительное время, так как это совершенно не мешает работе других устройств, зато позволяет передавать любые массивы информации с высокой скоростью, что также способствует повышению производительности комплекса. Заметим, что в МПВК с общей шиной передача информации массивами, т. е. занятие типа одной парой устройств на длительный отрезок времени, обычно допускается лишь в крайних случаях, так как это приводит к длительным простоям остальных устройств.

Кроме того, к достоинствам структуры с перекрестной коммутацией можно отнести простоту и унифицированность интерфейсов всех устройств, а также возможность разрешения всех конфликтов в коммутационной матрице. Важно отметить и то, что нарушение какой-то связи приводит не к выходу из строя всего комплекса, а лишь к отключению какого-либо устройства, т. е. надежность таких комплексов достаточно высока. Однако и организация МПВК с перекрестной коммутацией не свободна от недостатков.

Прежде всего – сложность наращивания ВК. Если в коммутационной матрице заранее не предусмотреть большого числа входов, то введение дополнительных устройств в комплекс потребует установки новой коммутационной матрицы. Существенным недостатком является и то, что коммутационная матрица при большом числе устройств в комплексе становится сложной, громоздкой и достаточно дорогостоящей. (Надо учитывать

то обстоятельство, что коммутационные матрицы строятся обычно на схемах, быстродействие которых существенно выше быстродействия схем и элементов основных устройств, – только при этом условии реализуются все преимущества коммутационной матрицы.) Это обстоятельство в значительной степени усложняет и удорожает комплексы.

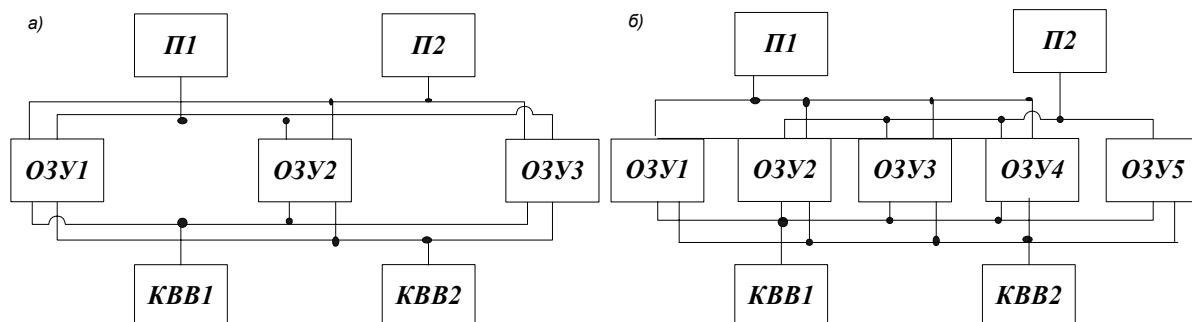


Рис. 2.9. МПК с многоходовым ОЗУ

Для того чтобы упростить и удешевить ВК, коммутацию устройств осуществляют с помощью двух и даже более коммутационных матриц. На рис. 2.8, б представлен МПК, включающий в себя две матрицы: КМЦУ – матрицу для центральных устройств (процессоров, ОЗУ и каналов ввода – вывода) и КМПУ – матрицу для периферийных устройств. Схемы последней могут иметь существенно меньшее быстродействие, чем схемы первой, да к тому же обе коммутационные матрицы будут значительно проще и дешевле, чем одна общая коммутационная матрица с высоким быстродействием.

Перекрестная коммутация довольно широко используется при построении ВК, в частности практически всех МПК фирмы «Барроуз» (в том числе и упомянутого выше комплекса D-825).

В МПК с многоходовыми ОЗУ все, что связано с коммутацией устройств, осуществляется в ОЗУ. В этом случае модули ОЗУ имеют число входов, равное числу устройств, которые к ним подключаются, т. е. для каждого устройства предусматривается свой вход в ОЗУ. Структура такого МПК показана на рис. 2.9, а. В отличие от ВК с перекрестной коммутацией, которые имеют централизованное коммутационное устройство, в МПК с многоходовыми ОЗУ средства коммутации распределены между несколькими устройствами. Такой способ организации МПК сохраняет все преимущества систем с перекрестной коммутацией, несколько упрощая при этом саму систему коммутации. Для наращивания системы должны быть предусмотрены дополнительные входы в ОЗУ. Правда, введение дополнительных модулей ОЗУ не вызывает затруднений.

В МПК с многоходовыми ОЗУ очень просто решается вопрос о выделении каждому процессору своей оперативной памяти, недоступной другим процессорам. Такая организация показана на рис. 2.9, б. Выделение индивидуальной памяти каждому процессору позволяет хранить в ней информацию, которая необходима только одному процессору: различные таблицы и данные, копии некоторых модулей операционной системы и др. Это позволяет избежать части конфликтов, которые неизбежно возникают при общей оперативной памяти. Кроме того, уменьшается вероятность искажения информации в ОЗУ другими процессорами. Однако такие БК имеют тот недостаток, что в случае выхода из строя какого-либо процессора доступ к его памяти затруднен и информация может быть переписана в другой модуль ОЗУ только через канал ввода – вывода и внешнее ЗУ, что требует много времени.

Приведенные три типа структурной организации исчерпывают существующие построения МПК, полностью удовлетворяющих тому определению, которое дано в начале этого параграфа. Такие МПК в литературе называют классическими или истинными МПК.

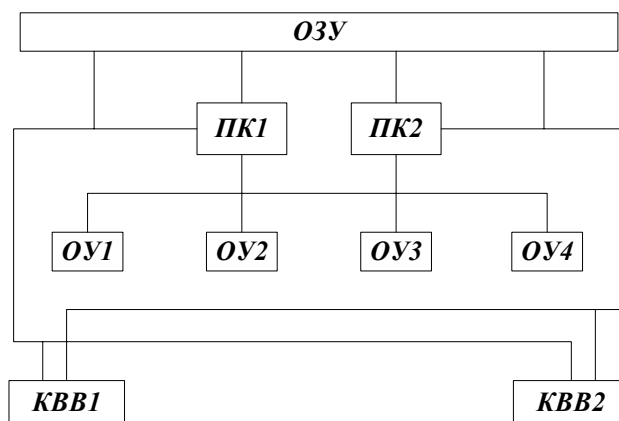


Рис. 2.10. Комплекс ЕС-1065.

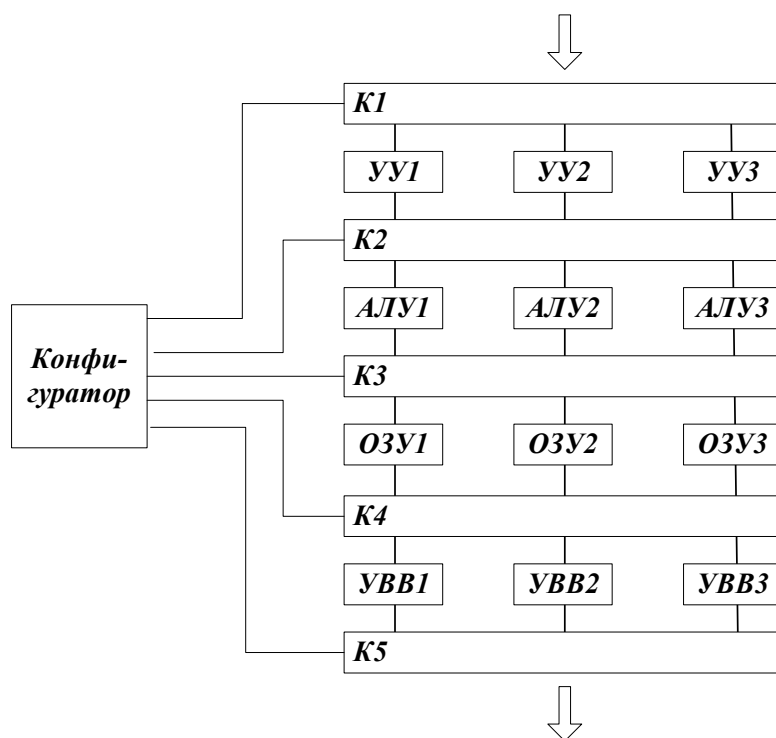


Рис. 2.11. Комплекс повышенной надежности

Вместе с тем нередко МПВК называют комплексы, лишь частично удовлетворяющие этому определению, например любые комплексы, в которых имеется несколько процессоров, а иногда даже матричные и конвейерные, которые (в отличие от классических МПВК) не относятся к комплексам с множественным потоком команд и множественным потоком данных. Кроме того, иногда к МПВК относят ЭВМ, имеющие не одно общее арифметическо-логическое устройство, а несколько операционных устройств (ОУ), каждое из которых выполняет определенную группу операций и может работать совершенно независимо от других. Таким образом, на первый взгляд может показаться, что такой комплекс имеет несколько процессоров, каждый из которых связан с общим ОЗУ. Однако все ОУ работают под управлением одной программы.

Но вот ЭВМ ЕС-1065, в которой применяются множественные ОУ, с полным правом можно называть МПВК, так как кроме этих ОУ имеется два процессора команд (ПК), причем каждый может работать по собственной программе (рис. 2.10). В этом комплексе налицо все элементы: несколько независимых процессоров, работающих с общедоступными ОЗУ, имеющих общие периферийные устройства и работающих под управлением общей операционной системы. По типу структурной организации комплекс ЕС-1065 может быть отнесен к МПВК с многовходовыми ОЗУ.

Нередко к МПВК относят комплексы, являющиеся по существу ММВК, но в которых для достижения более высокой надежности резервирование осуществляется не «помашинно», а по устройствам. Структурная схема одного из таких ВК приведена на рис. 2.11. Комплекс первоначально работает как трехмашинный со сравнением результатов на выходе. Результат используется по методу «голосования». При несовпадении результата с помощью системы контроля определяется и отключается неисправное устройство. Эти функции выполняются конфигуратором, контролирующим состояние комплекса. Такой комплекс обладает высокой надежностью, однако, как видим, несмотря на наличие трех процессоров, его нельзя назвать многопроцессорным.

2.4. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

Хотя и ММВК и МПВК относятся к комплексам с множественным потоком команд и множественным потоком данных, организация процессов обработки в них существенно различна.

Многомашинный ВК в наибольшей степени приспособлен для решения потока не связанных между собой задач, т. е. такой совокупности задач, когда исходные данные, а также окончательные или промежуточные результаты решения одной задачи не используются при решении других задач. Строго говоря, в ММВК можно достаточно эффективно решать и слабосвязанные задачи, т. е. задачи, при решении которых объем взаимно передаваемой информации относительно невелик. При этом нетрудно заметить, что чем больше объем взаимно передаваемой информации, тем в большей степени снижается производительность системы, так как часть времени приходится расходовать на обмен информацией.

Анализ показывает, что решение одной задачи несколькими машинами в составе ММВК может быть оправданным только при наличии связи между ними через общее ОЗУ. Необходимо отметить, что в этом случае трудность решения определяется не столько потерями производительности из-за обмена информацией, сколько сложностью синхронизации этапов решения, а, следовательно, действия ЭВМ. По этим причинам ММВК в основном используются для повышения надежности и увеличения производительности при решении потока независимых или слабосвязанных задач. Заметим, что производительность растет практически линейно с увеличением числа ЭВМ.

С учетом высказанных соображений можно дать оценку сложности операционных систем ЭВМ, входящих в состав ММВК.

Как правило, ОС в ММВК сравнительно мало отличаются от ОС одиночных ЭВМ, работающих в мультипрограммном режиме обработки информации: просто ОС одиночной ЭВМ дополняется компонентами, обеспечивающими выполнение функций, связанных с комплектованием. Обычно ОС в ММВК содержат следующие дополнительные программные модули: 1) обеспечения обмена информацией между ЭВМ; 2) взаимного контроля состояния ЭВМ; 3) проведения регламентных работ; 4) взаимодействия ВК с оператором.

Программы обмена информацией могут быть различными по сложности в зависимости от характера межмашинных связей.

Как уже отмечалось, наиболее сложные программы обмена имеют место в ВК, которые предназначены для решения связанных задач или одной общей задачи. Кроме того, программы обмена обеспечивают передачу необходимой информации резервной ЭВМ и ВК с резервированием.

Программы взаимного контроля состояния ЭВМ необходимы в тех случаях, когда ВК предназначен для обеспечения повышенной надежности. Взаимный контроль состояния совершенно необходим, когда предусматривается автоматический или автоматизированный переход с основной ЭВМ на резервную, причем если такое переключение осуществляется по командам оператора, используется программа, обеспечивающая взаимодействие ВК с оператором.

В режиме длительного функционирования ВК возникает необходимость проведения е определенной периодичностью регламентных работ. При этом поочередно все машины

выполняют функции основной и резервной ЭВМ. Для обеспечения такого перехода без перерывов в работе и служит программа проведения регламентных работ.

Хотя в некоторых случаях все эти программы могут быть достаточно сложными, однако их общий объем сравнительно невелик – редко превышает 10% объема ОС.

Многопроцессорные ВК имеют существенно большие возможности и большую гибкость, чем ММВК, как в отношении повышения надежности, так и в отношении организации параллельных вычислений. Эти возможности определяются тем, что все процессоры могут работать с общей памятью, иметь в своем непосредственном распоряжении нее периферийные устройства (и том числе и ВЗУ) и управляться единой ОС.

Организация процесса обработки информации может быть различной в зависимости от тех целей, которые преследует использование МПВК.

Нетрудно заметить, что МПВК способны делать все то, что делает ММВК: решать независимые и слабосвязанные задачи. Но МПВК с существенно большей эффективностью могут решить сильносвязанные задачи, и, более того, МПВК способны эффективно решать одну задачу с участием всех процессоров. При этом благодаря использованию общей памяти всеми процессорами не требуется никакой передачи информации.

При решении одной задачи на нескольких процессорах можно в полной мере использовать параллелизм независимых ветвей. Таким образом, если ММВК позволяют достигать высокой производительности при решении большого потока задач, то МПВК позволяет это сделать даже при решении одной задачи, что чрезвычайно важно. Кроме того, МПВК обладают колоссальными возможностями для обеспечения высокой надежности: выход из строя любого процессора не приводит к отказу комплекса, а лишь снижает его производительность, так как исправные процессоры могут практически в любой момент подхватывать и продолжать решение задачи, которая выполнялась отказавшим процессором. Это же относится и к любым другим устройствам комплекса, т.е. пока в действии хотя бы одно устройство данного типа, комплекс продолжает работу, только с меньшей эффективностью.

Наконец, весьма важным является то, что МПВК позволяет повысить коэффициент использования оборудования. Хорошо известно, что в ЭВМ даже в режиме мультипрограммной обработки информации почти все устройства значительную часть времени простаивают, так как процессор не в состоянии их загрузить.

Однако использование всех богатых возможностей МПВК является не простой задачей, так как операционные системы, на которые падает основная тяжесть организации вычислительных процессов, оказываются очень сложными. Операционные системы можно упрощать, если отказываться от тех или иных возможностей ВК. На это нередко идут разработчики ВК, получая взамен урезанных возможностей более простую ОС.

В настоящее время известны три типа организации вычислительного процесса в МПВК и соответственно три типа функционирования ОС: 1) ведущий – ведомый; 2) раздельное выполнение заданий в каждом процессоре; 3) симметричная, или однородная, обработка информации всеми процессорами.

Организация работы МПВК по принципу «ведущий – ведомый» является наиболее простой. Один из процессоров управляет работой всех остальных, т. е. берет на себя функции распределения задач и ресурсов, организует необходимую передачу информации, производит переназначение задач в случае выхода из строя какого-либо процессора, отключение неисправных устройств и т. д. Остальные процессоры выполняют только функции непосредственной обработки: являются исполнительными процессорами.

Операционная система такого МПВК получается также достаточно простой, вследствие того, что супервизорные функции выполняет один процессор, конфликты из-за ресурсов отсутствуют, да и число конфликтов от других причин сводится к минимуму (так как организация процесса обработки централизована и конфликтные ситуации могут быть предотвращены при распределении задач и ресурсов между процессорами). Кроме того,

каждый процессор располагает своим набором различных таблиц, что упрощает организацию параллельных процессов.

Ведущий процессор может ничем не отличаться от остальных – просто его функции выполняет любой назначенный оператор процессор. В случае отказа ведущего процессора оператор может произвести переназначение – сделать ведущим другой процессор.

Иногда, учитывая специфичность выполняемых ведущим процессором функций, его делают специализированным, настроенным на выполнение узкого круга операций, добиваясь за счет этого высокой производительности. Однако такое решение имеет тот недостаток, что при отказе ведущего процессора из строя выходит комплекс. К недостаткам организации по типу «ведущий – ведомый» следует отнести малую ее гибкость. При большом потоке коротких задач ведущий процессор может не справляться с распределением задач между исполнительными процессорами, что неизбежно приводит к простоям некоторых из них. По той же причине не очень эффективно работает комплекс при решении на нескольких процессорах одной задачи: в этом случае на ведущий процессор ложится большая работа по синхронизации процесса решения частей задачи. Многопроцессорный ВК с организацией по принципу «ведущий – ведомый» в наибольшей степени подходит для создания специализированных систем, в которых недостатки такой организации могут быть заранее учтены и их влияние сведено к минимуму. Но еще раз подчеркнем, что недостатки в значительной степени искупаются простотой операционной системы.

Значительно большей универсальностью обладает организация обработки информации с отдельным выполнением заданий в каждом процессоре. В этом случае все процессоры равноправны и каждый выполняет как супервизорные, так и исполнительные функции, но процессоры не свободны в выборе задач: каждому процессору заранее определяется набор задач. Практически заранее распределяются и ресурсы между процессорами. Каждая задача полностью выполняется одним процессором. При отказах какого-либо процессора комплекс сохраняет работоспособность, но повторный запуск программ, выполняемых отказавшими процессорами, представляет определенные трудности, из-за того что, во-первых, это может сделать только оператор, во-вторых, должен быть произведен перерасчет загрузки процессоров. При этой организации каждый процессор обслуживает только самого себя, удовлетворяя лишь свои потребности. Так как операционная система общая для всех процессоров, то супервизорные программы должны обладать повторной входимостью, т. е. несколько процессоров должны иметь возможность работать с одной и той же программой ОС. Чтобы избежать такого требования к ОС, иногда используется другой способ – каждый процессор имеет свою копию ОС.

В МПВК с отдельным выполнением заданий возникает мало конфликтов, так как ресурсы распределены заранее и каждый процессор имеет собственный набор таблиц. При этой организации трудно обеспечить равномерную загрузку процессоров, да и устройств, так как практически невозможно все просчитать заранее и тем более сложно сделать это в процессе работы. Поэтому могут быть простои части оборудования. Такая организация обработки информации в МПВК делает его похожим на ММВК. Однако есть и существенные отличия: прежде всего ресурсы используются гораздо эффективнее, так как общее число устройств может быть значительно меньше, чем в ММВК; кроме того, при необходимости обеспечить высокую надежность решения определенных задач резервирование может быть осуществлено проще и эффективнее за счет того, что все процессоры используют общие ресурсы. И опять-таки значительно компенсирует недостатки этого способа организации вычислений относительная простота системы.

В наибольшей степени все преимущества МПВК проявляются при симметричной, или однородной, обработке. Все процессоры в этом случае имеют максимальную самостоятельность. В комплексе не существует предварительного распределения заданий между процессорами – каждый из них при освобождении от решения предыдущей задачи выбирает себе новую из общего списка (или очереди). Нет и предварительного распределения ресурсов – процессоры сами набирают их, исходя из потребностей решаемой

задачи. Все процессоры могут участвовать в решении одной задачи, если отдельные ее части включаются в общий список заданий. В комплексе автоматически отключается неисправное оборудование, при этом ВК продолжает работу с несколько пониженной производительностью. Таким образом, отказа ВК нет до тех пор, пока не исчерпано все однотипное оборудование. Комплекс характеризуется хорошей загрузкой всех устройств, что позволяет достигать высокой эффективности.

При симметричной обработке на каждый процессор могут возлагаться и общесистемные, и супервизорные, и исполнительные функции. Однако такая организация не свободна и от некоторых недостатков. Основные из них – большое число конфликтных ситуаций и сложность ОС. Последний недостаток нередко заставляет разработчика МПВК идти на компромиссные решения, вводя ограничения, о которых говорилось раньше.

В чем же сложность ОС многопроцессорных комплексов? На первый взгляд кажется, что ОС МПВК мало чем отличается от ОС мультипрограммной однопроцессорной ЭВМ: та же структура, тот же состав – планировщик, диспетчер, супервизор ввода – вывода и т. д. Однако детальное рассмотрение показывает, что по существу различие весьма глубокое. Чем же это объяснить? Прежде всего, не надо забывать, что в однопроцессорной ЭВМ при мультипрограммной обработке имеет место лишь совмещение во времени этапов решения задачи. Параллельная работа разных устройств позволяет одновременно вести обработку информации по нескольким программам; но при этом собственно обработка в процессоре производится в каждый данный момент времени только по одной программе. Таким образом, работа по нескольким программам осуществляется лишь за счет совмещения во времени работы процессора по одной программе с работой устройств ввода–вывода по другим программам.

В МПВК имеет место не только такое совмещение, но и одновременная обработка информации несколькими процессорами по нескольким программам. Это совершенно другое качество, которое приводит к усложнению всех процессов. Действительно, рассмотрение, к примеру, работы ОЗУ в однопроцессорной ЭВМ и в МПВК показывает, что если в первом случае конфликты возникают только при одновременном обращении процессора и КВВ к одному модулю памяти и легко разрешаются благодаря приоритету КВВ, то в МПВК конфликты возникают и при одновременном обращении нескольких процессоров к одному модулю памяти, не говоря уже о том, что и число КВВ существенно больше. Разумеется, организация очереди и решение всех конфликтов требует соответствующих программ в ОС МПВК. В однопроцессорной ЭВМ не может быть конфликтов при обращении к стандартным программам да и конфликты из-за программных ресурсов маловероятны, в то время как в МПВК это совершенно нормальное явление. Кроме того, при симметричной обработке информации супервизорные программы, как уже отмечалось, должны иметь повторную входимость, что также усложняет ОС. Таким образом, одновременное выполнение нескольких программ предъявляет к ОС МПВК серьезные дополнительные (по сравнению с ОС однопроцессорной ЭВМ) требования.

Второй важный момент заключается в том, что ОС в однопроцессорной ЭВМ работает с фиксированной конфигурацией ЭВМ. Операционная система МПВК должна не только обеспечивать работу с различными конфигурациями, а более того – осуществлять автоматическую реконфигурацию комплекса (при симметричной обработке). Только при этом условии может быть достигнута наибольшая надежность, которую способен обеспечить данный комплекс. Реконфигурация всегда влечет за собой перераспределение ресурсов. Одновременная работа нескольких процессоров и необходимость реконфигурации комплекса значительно усложняют ОС МПВК.

Рассмотрим некоторые особенности ОС МПВК, которые отличают ее от ОС однопроцессорной системы.

При управлении ОЗУ на ОС МПВК возлагаются в основном те же задачи, что и на ОС однопроцессорной ЭВМ. Но есть одна проблема, которая отсутствует в однопроцессорной ЭВМ. Дело в том, что для каждого процессора нужна своя оперативная память для хранения

некоторых собственных таблиц, слов состояния, прерываний, программ восстановления и др. Адресация этой памяти имеет определенную специфику. Чаще всего в МПВК для этих целей существует так и называемая *префиксация адресов*.

Сущность префиксации заключается в том, что адреса ячеек зоны памяти, которую использует только данный процессор, формируются с использованием содержимого регистра префикса, который задает необходимое начальное смещение этой зоны. В § 2.6 при описании МПВК на базе ЕС ЭВМ префиксация адресов описывается более подробно.

Выше упоминалось о том, что в МПВК существенно больше конфликтов при обращении к ОЗУ, и это требует соответствующей организации ОС. Заметим, что для МПВК сокращение цикла работы ОЗУ имеет большее значение, чем в однопроцессорной ЭВМ, так как уменьшает простои процессоров из-за конфликтов.

Отметим еще одну особенность МПВК. Известно, что в однопроцессорных ЭВМ значительный эффект дает чередование, или расслоение адресов. В МПВК это может не дать никакого эффекта или этот эффект будет сильно ослаблен, поскольку вероятность того, что разные процессоры и разные КВВ будут обращаться к соседним ячейкам памяти, весьма мала. Наличие сверхоперативных ЗУ у каждого процессора уменьшает число конфликтов, однако возникают определенные трудности с повторным запуском задач при выходе из строя процессора.

Планирование и диспетчеризация в МПВК существенно сложнее, чем в однопроцессорных системах. Пожалуй, основная проблема – определение задач, и в особенности частей одной задачи, которые могут решаться параллельно. Этому уделяется большое внимание. Разработаны и широко используются языки программирования, в которых предусматриваются средства для организации параллельной обработки задач. Достаточно сложна и задача синхронизации параллельных процессов. Для этой цели часто используются так называемые семафоры, позволяющие приостановить выполнение процесса до удовлетворения определенных условий. Семафор – это обычно переменная величина, представляющая собой целое число, которое уменьшается или увеличивается на единицу при определенных событиях.

Второй сложной и важной проблемой в планировании является равномерная загрузка процессоров и периферийных устройств. Задача эта не такая простая, как может показаться на первый взгляд. Дело в том, что равномерности загрузки можно добиться только при достаточно точном прогнозировании времени решения каждой задачи.

Один из неприятных моментов в МПВК – «тупиковые» ситуации. «Тупики» возникают, когда для решения двух (или более) активных, т. е. уже находящихся в стадии решения, задач требуются одни и те же ресурсы (УВВ, наборы данных, программы, ОЗУ), а приоритет одинаков. «Тупики» преодолеваются с помощью ОС. Существует несколько методов, но, как правило, все они приводят к тому, что одну задачу, а иногда и более, приходится снимать, т. е. переводить в режим ожидания решения.

Говоря о планировании и диспетчеризации, нельзя не сказать о важности защиты информации. Эта проблема существует и в однопроцессорной ЭВМ, и там она решается, как правило, аппаратными средствами. В МПВК этого бывает недостаточно, так как системные таблицы, некоторые наборы данных могут использоваться несколькими процессорами, т. е. все они имеют право обращаться к этой информации и аппаратная защита разрешает обращение. Возможные действия одного из процессоров, не согласованные с работой других, могут привести к искажениям информации. Эту проблему приходится решать также с помощью ОС.

Выше отмечалось, что на ОС МПВК возлагается автоматическая реконфигурация комплекса при отказах тех или иных устройств, благодаря чему эти отказы не выводят из строя всего комплект. Но ОС МПВК должна учитывать и то обстоятельство, что некоторые отказы могут вызвать искажение информации. При этом процесс нарушается и никакая реконфигурация комплекса не в состоянии исправить это. Чтобы этого не случилось, обычно

все системные таблицы и таблицы управления заданиями дублируются, что даст возможность при их искажении продолжать процесс решения.

2.5. КОМПЛЕКСЫ НА ОСНОВЕ МИКРО-ЭВМ И МИКРОПРОЦЕССОРОВ

Бурное развитие микропроцессорной техники явилось новым стимулом для развития ВК, так как преимущества, которые дает эта техника, не могли остаться без внимания со стороны разработчиков ВК. А преимущества, как известно, значительные – высокая надежность, низкая стоимость, малые габариты, масса, потребляемая мощность, возможность использования в качестве встроенных элементов в любом оборудовании. Микропроцессоры обеспечили создание дешевых малогабаритных микро-ЭВМ с достаточно высокими характеристиками, процессоров, контроллеров периферийных устройств. Поэтому появилось естественное желание использовать все эти качества в максимальной степени. Правда, современные микро-ЭВМ и микропроцессоры имеют сравнительно невысокую производительность, однако их положительные качества открывают возможность для создания комплексов более крупных, т. е. из большего числа ЭВМ или устройств. Действительно, вряд ли целесообразно создавать ММВК, например из 10–15 ЭВМ типа ЕС или даже СМ, настолько громоздким и сложным был бы такой комплекс. Но эта идея не вызывает никакого протеста, когда речь идет о построении комплекса из 10 и более одно- или двухплатных микро-ЭВМ. Даже при относительно низкой производительности микро-ЭВМ производительность таких комплексов может составлять миллионы операций в секунду. Двойное или тройное резервирование аппаратуры обеспечивает очень высокую надежность. Если же учесть, что микропроцессорные ВК легко встраиваются в любое оборудование, агрегаты, транспортные средства, становится ясным, насколько актуальны такие комплексы.

Вместе с тем нельзя не учитывать, что при использовании в одном ВК большого числа микропроцессорных средств возникает целый ряд проблем, в первую очередь сложность коммутации, большое число конфликтов, сложность организации вычислительных процессов. Выше уже было указано, что эти проблемы существуют и в ВК с малым числом ЭВМ или процессоров, однако в более крупном комплексе проблемы обостряются.

Стремление преодолеть возникающие трудности привело к поиску новых способов построения комплексов на основе микро-ЭВМ и микропроцессоров. Все существующее в настоящее время многообразие комплексов можно разделить на три большие группы. 1) первой группе относятся комплексы, выполняемые по классическим схемам многомашинных или многопроцессорных ВК, с теми способами связи и взаимодействия элементов, которые уже были описаны. Во вторую группу входят комплексы, построенные на микро-ЭВМ, память которых доступна для всех ЭВМ комплекса, – так называемые ВК с общедоступной памятью. Третья группа характеризуется наличием общего поля оперативной памяти.

Можно определить и четвертую группу комплексов, которые трудно классифицировать по какому-либо признаку и которые имеют специфические особенности либо в способах коммутации устройств, либо в способах и средствах организации вычислительного процесса.

Остановимся более подробно на каждой группе с целью выявления их возможностей и особенностей.

Следует заметить, что появление микропроцессоров поначалу породило большие надежды в отношении классических МПВК с большим числом процессоров, так как снимались серьезные ограничения, связанные со сложностью, громоздкостью и высокой стоимостью комплексов. Делались попытки создания МПВК, включающих в себя десятки и даже тысячи процессоров. Характерным в этом отношении является комплекс «Гиперкуб», содержащий 10000 процессоров [5]. Комплекс, как и другие подобные ему, не оправдал возлагавшихся на него надежд из-за сложности построения, организации вычислительных процессов и трудностей программирования. Это заметно охладило пыл энтузиастов МПВК,

так как стало ясным, что основные трудности создания МПВК остались и при увеличении числа процессоров только возрастают. Поэтому дальнейшего развития ВК с большим числом процессоров не получили, но комплексы с ограниченным числом (два – четыре) процессоров используются достаточно часто.

Многомашинные ВК на базе микро-ЭВМ создавались первоначально по классическим схемам. Однако это не давало должного эффекта при большом числе микро-ЭВМ. Получались комплексы, которые развивали достаточно высокую производительность только при решении больших потоков сравнительно простых задач. Не удавалось добиться высокой производительности при решении сложных задач за счет организации параллельной обработки информации. Это и привело к поискам новых структур. Ясно было, что надо придавать многомашинным ВК некоторые свойства, присущие многопроцессорным комплексам. В основном вели поиск по пути объединения всех или части ресурсов комплекса. Многочисленные полученные варианты, как уже отмечалось, можно объединить в группы. Заметим только, что все варианты имеют общий характерный признак: это многомашинные комплексы с включением некоторых особенностей МПВК.

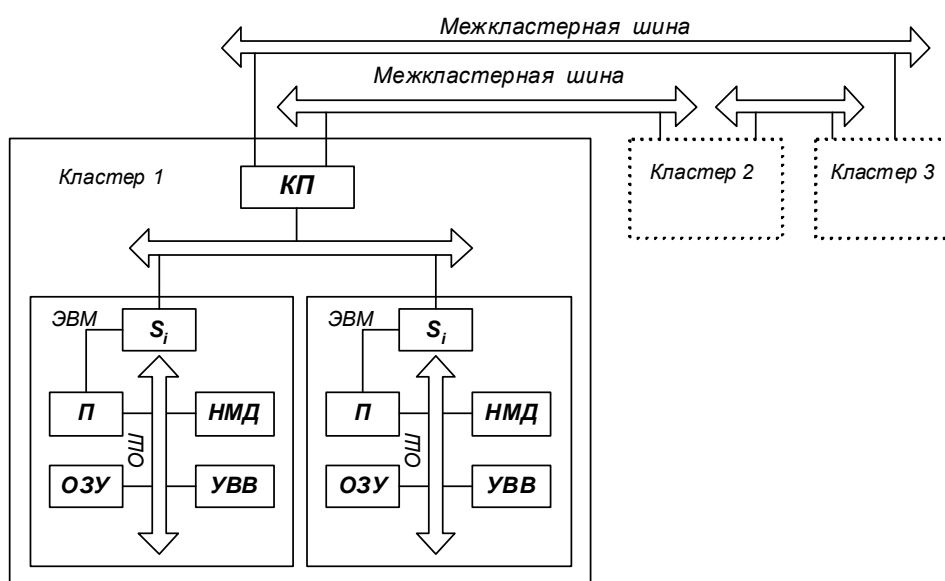


Рис. 2.12. Комплекс типа Cm*

Идея комплексов с общедоступной памятью очень проста – ОЗУ и ВЗУ каждой ЭВМ доступны для других ЭВМ. С этой целью предусматриваются соответствующие связи, а система адресации позволяет рассматривать все ЗУ как единое целое. При этом каждая ЭВМ имеет все необходимые ресурсы для решения задач и управляется своей операционной системой, т. е. комплекс остается многомашинным. Однако общедоступная память дает новое качество: возможность использовать общие базы данных, решать на нескольких ЭВМ одну, общую, задачу, затрачивая на обмен информацией сравнительно малое время. Следует заметить, что современные ОЗУ на интегральных схемах обладают достаточно высоким быстродействием, в то время как быстродействие процессоров относительно невелико, поэтому замедление работы ЭВМ при обращении к памяти другой ЭВМ оказывается не очень значительным. При решении одной задачи на нескольких ЭВМ, как и в МПВК, остается проблема синхронизации этапов решения.

Пожалуй наиболее характерным в этой группе является комплекс Cm* (рис. 2.12), разработанный университетом Карнеги (США). Основным элементом комплекса является микро-ЭВМ LSI-11 фирмы DEC, включающая в себя процессор, ОЗУ, ВЗУ на сменных магнитных дисках, устройства для работы оператора. Все устройства объединены общей шиной. Несколько таких ЭВМ (до 14) связываются через так называемую шину отображения с помощью местных (локальных) переключателей S_i . Один переключатель объединяет общую шину LSI-11, процессор и шину

отображения. Группа ЭВМ, связанных шиной отображения, образует так называемый кластер. В комплексе может быть несколько кластеров, которые соединяются между собой через межкластерные шины. Взаимодействие кластеров обеспечивается быстродействующим коммутационным процессором (КП) с микропрограммным управлением. Этот процессор одновременно управляет и шиной отображения. На рисунке представлена схема ВК из трех кластеров. Каждый процессор имеет доступ ко всем ЗУ системы. Система адресации позволяет образовать 28-разрядное виртуальное адресное пространство, которое разбито на сегменты по 4096 байт, т. е. имеет 2^{16} сегментов. Естественно, в этом комплексе доступ к различным ЗУ осуществляется с разной задержкой: минимальной – при обращении к собственной памяти, средней – при обращении к памяти другой ЭВМ, входящей в тот же кластер, и максимальной – при обращении в ЗУ ЭВМ других кластеров.

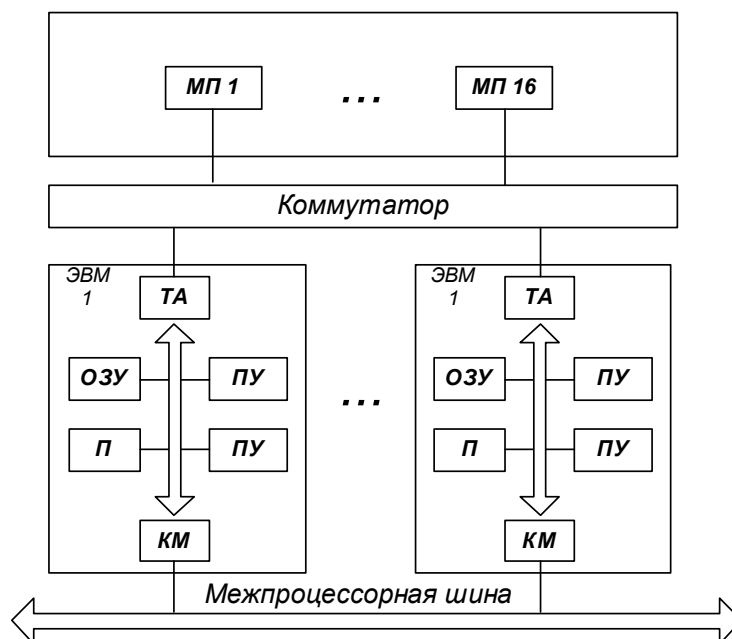


Рис. 2.13. Комплекс типа Сттпр

Принцип построения систем с общим полем оперативной памяти заключается в том, что несколько ЭВМ, каждая из которых имеет собственное ОЗУ, кроме того, имеют возможность обращаться к общему ОЗУ, емкость которого значительно больше не только емкости ОЗУ одной ЭВМ, но и суммарной емкости всех ОЗУ. Такое построение системы в максимальной степени приближает многомашинный ВК к многопроцессорному, так как наличие общего оперативного ЗУ обеспечивает комплексу большинство свойств, характерных для МПВК, в том числе и возможность параллельной работы при решении одной большой задачи. При этом каждая ЭВМ работает под управлением своей, достаточно простой ОС.

Характерным представителем этой группы является комплекс типа Сттпр (рис. 2.13). Этот ВК построен на основе широко известных мини-ЭВМ PDP-11/40 фирмы DEC. В комплексе может быть до 16 таких ЭВМ. С помощью матричного коммутатора (16X16) к любой из этих ЭВМ может подключаться один из 16 модулей памяти (МП), создающих таким образом общее для процессоров всех ЭВМ поле памяти. Каждый модуль памяти имеет емкость 1 млн. слов (2 Мбайт). Так как адресное поле ЭВМ составляет 18 разрядов, а общее адресное пространство равно 32 Мбайт, обращение процессоров к общей памяти осуществляется через специальный транслятор адреса (ТА), который работает с достаточно высокой скоростью: задержка не превышает 250 нс.

Кроме связи ЭВМ через общую память имеется связь посредством общей межпроцессорной шины, что обеспечивает взаимные прерывания с тремя уровнями, синхронизацию процессоров, а также

выполнение некоторых функций управления (каждый процессор может останавливать и запускать любой другой процессор).

Процессоры PDP-11/40, которые используются в этом комплексе, несколько отличаются от серийных. В них внесены изменения, обеспечивающие запись в управляющую память и возможность ее расширения. Пользователю запрещено применять некоторые команды (HALT, WAIT и др.). Операционная система также претерпела изменения. Емкость собственного ОЗУ каждой ЭВМ составляет 4 кслов (8 кбайт). Такой же выбрана емкость страницы в общем поле памяти. Конфликты, которые возникают при одновременном обращении двух или более процессоров и одному и тому же модулю памяти, раз решаются в самих модулях, где организуется очередь запросов.

Как отмечалось, ВК типа $Stmp$ максимально приближаются к микропроцессорным и при малой емкости памяти собственных ОЗУ приобретают значительную часть недостатков МПВК, в частности большое число конфликтов из-за памяти, сложность ОС. По-видимому, лучшие результаты можно получить, увеличивая собственную оперативную память ЭВМ, образующих комплекс. Возможности ЭВМ в этом варианте резко возрастают, существенно уменьшается число конфликтов из-за памяти, ОС приближается по своему характеру к ОС ММВК и становится проще. Вместе с тем сохраняется и ценное качество МПВК: возможность эффективного взаимодействия всех или части процессоров при решении общей задачи.

В отношении надежности ВК типа St^* и $Stmp$ безусловно эффективны, так как оба позволяют осуществлять резервирование всех ресурсов и допускают в значительных пределах реконфигурацию при выходе из строя того или иного устройства. В простейшем случае возможно простое отключение неисправной ЭВМ с передачей выполняемых ею функций другим ЭВМ.

Рассматривая организацию обработки информации в таких ВК, можно отметить, что принципиально и в том и в другом комплексе возможны все три способа функционирования: «ведущий – ведомый», раздельное выполнение заданий каждым процессором, симметричная обработка. Однако для комплексов типа St^* больше подходит организация по типу «ведущий – ведомый», а организация симметричной обработки сильно затруднена. Для ВК типа $Stmp$ одинаково приемлемы все три способа.

2.6. РЕАЛИЗАЦИЯ КОМПЛЕКСОВ

В отечественной вычислительной технике в настоящее время можно выделить три основных направления развития [5]:

1. Единая система ЭВМ социалистических стран (ЕС ЭВМ).
2. Система малых ЭВМ социалистических стран (СМ ЭВМ).
3. Семейство высокопроизводительных комплексов «Эльбрус».

Кроме этих направлений, разумеется, существуют и другие, однако эти три определяют в основном технический уровень и парк вычислительных машин в стране.

Концепции, положенные в основу построения ЕС и СМ ЭВМ, с самого начала предусматривали создание семейства не только однопроцессорных ЭВМ, но и вычислительных комплексов на основе технических и программных средств этих ЭВМ. Семейство «Эльбрус», например, реализуется только в виде многопроцессорных вычислительных комплексов.

Представляется целесообразным рассмотреть возможности и принципы организации вычислительных комплексов для всех направлений.

ВК на базе ЕС ЭЦМ. Единая система ЭВМ представляет собой семейство программно-совместимых вычислительных машин, предназначенных для решения широкого класса научно-технических, экономических, управленческих и других задач. ЕС ЭВМ разрабатывается и производится совместно странами социалистического содружества: НРБ, ВНР, ГДР, ПНР, СРР, СССР и ЧССР. Разработка начата в 1969 г., а серийное производство ведется с 1971 г. Научные и производственные коллективы всех стран работают по единому плану. В настоящее время ЭВМ Единой системы составляют основу парка ЭВМ социалистических стран. Разработка ЕС ЭВМ по объему капиталовложений, по числу организаций-соисполнителей и по составу разрабатываемых средств является крупнейшим проектом в области вычислительной техники в Советском Союзе и социалистических странах. Объединение усилий для решения общей задачи позволило сделать большой шаг вперед в развитии научно-технической и производственной базы вычислительной техники стран – участниц разработки.

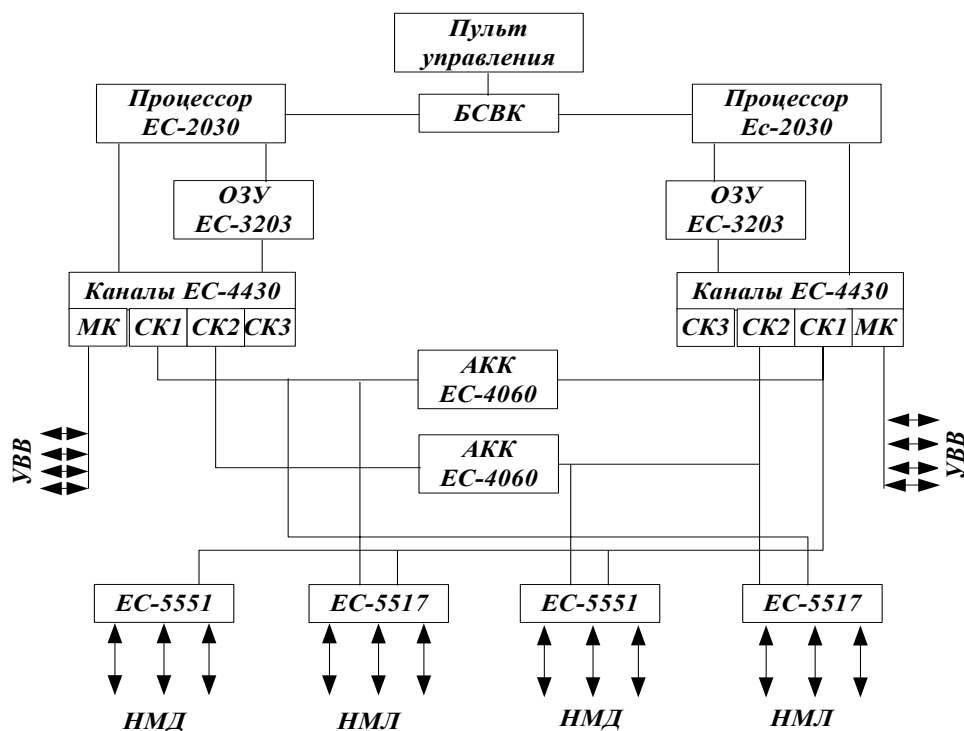


Рис. 2.14. Комплекс ВК-1010

В рамках ЕС ЭВМ создавались двухмашинные и двухпроцессорные комплексы, которые позволяют повысить производительность и надежность и вместе с тем отличаются относительной простотой, как технической реализации, так и программного обеспечения. Это вовсе не означает, что имеющиеся средства не позволяют создавать, скажем, трех- или четырехмашинные и трех- или четырехпроцессорные комплексы. Дело в том, что такие комплексы будут существенно сложнее и их технико-экономическая эффективность может оказаться недостаточно высокой.

Первым двухмашинным комплексом в составе ЕС был комплекс ВК-1010, построенный на базе ЭВМ ЕС-1030: В этом комплексе были использованы почти все существующие способы организации связи между ЭВМ: прямое управление, адаптер канал – канал, связь на уровне общих ВЗУ. Таким образом, в комплексе ВК-1010 не было лишь связи между ЭВМ через общую оперативную память. Структурная схема ВК-1010 (рис. 2.14) осталась практически без изменения и для последующих двухмашинных ВК созданных на базе ЕС ЭВМ-1 и ЕС ЭВМ-2: ВК-1033 (из двух ЕС-1033), ВК-2Р-35 (ЕС-1035), ВК-2Р-45 (ЕС-1045), ВК-2Р-60 (ЕС-1060) [5]. Эту схему можно считать типовой для ЕС ЭВМ. Рассмотрим средства, используемые в ЕС ЭВМ для создания двухмашинных комплексов.

Обмен управляющей и синхронизирующей информацией между ЭВМ осуществляется с помощью средств прямого управления, к которым относятся стандартный интерфейс прямого управления, специальные команды ПРЯМАЯ ЗАПИСЬ и ПРЯМОЕ ЧТЕНИЕ и механизм внешних прерываний. Эти средства позволяют осуществлять быструю связь между центральными процессорами и обычно используются для передачи небольших объемов информации. Через канал прямого управления к ВК подключается блок состояния вычислительного комплекса (БСВК) и пульт управления ВК, которые вместе образуют устройство управления вычислительным комплексом. Блок состояния ВК, включает в себя блок управления и регистры состояния, которые определяют режим работы ВК. Запись в эти регистры осуществляется либо оператором с пульта, либо по команде ПРЯМАЯ ЗАПИСЬ. С помощью команды ПРЯМОЕ ЧТЕНИЕ содержимое этих регистров может быть переписано в оперативную память.

Интерфейс прямого управления включает в себя: восемь входных и восемь выходных линий информации, четыре линии синхронизации, две линии внешних сигналов, линии

записи и чтения. Блок состояния совместно с пультом управления ВК позволяет осуществлять ручное переключение ВК и ЭВМ в требуемый режим работы; начальную загрузку программ; внешнее прерывание; включение и выключение питания. Кроме того, на пульте управления осуществляется индикация состояний ЭВМ, а также необходимая сигнализация об аварийных состояниях.

Адаптеры канал–канал позволяют производить обмен большого объема информации между ЭВМ; АКК работает в монопольном режиме с высокой пропускной способностью (примерно 1 Мбайт/с). Для каждой ЭВМ адаптер является как бы устройством, которое выбирается каналом: точно так же реагирует на все запросы канала, принимает и расшифровывает команды канала – с той только разницей, что команды и сигналы используются не для управления периферийным устройством, а для передачи информации между каналами и синхронизации их работы. В соответствии с функциями АКК в его структуру входят два блока управления обменом, связанных непосредственно с помощью нескольких сигнальных линий, а также через общий буферный регистр.

Третье средство комплексирования – связь ЭВМ через ВЗУ: накопители на магнитных лентах (НМЛ) и магнитных дисках (НМД). Все устройства управления НМЛ и НМД имеют двухканальные переключатели (ДКП), которые позволяют подключать УУ либо к двум каналам, либо к каналам двух ЭВМ. В первом случае ДКП обеспечивает доступ к ВЗУ через два канала, повышая тем самым надежность ЭВМ. Во втором ДКП позволяет организовать общее поле внешней памяти на НМЛ и НМД. Резервирование НМЛ к НМД тем или другим каналом осуществляется по команде ЗАРЕЗЕРВИРОВАТЬ УСТРОЙСТВО, которую выдает канал, однако есть некоторая разница в резервировании НМЛ и НМД. в первом случае канал резервирует все устройство управления, т. е. группу НМЛ, во втором – определенный накопитель. Освобождение накопителя осуществляется по команде ОСВОБОДИТЬ УСТРОЙСТВО.

Рассмотрим режимы функционирования ММВК на примере двухмашинного ВК. При исправности обеих ЭВМ возможны три режима работы. Первый: обе машины параллельно решают одни и те же задачи, однако используются результаты, выдаваемые только одной из них, которая считается основной. В случае выхода ее из строя происходит немедленное переключение на вторую (резервную) ЭВМ. Второй режим: обе ЭВМ работают как две независимые ЭВМ, каждая из которых решает свои задачи. В случае отказа одной из них вторая принимает на себя ее нагрузку. При этом либо увеличивается время пребывания задач в ЭВМ, либо решаются только самые необходимые задачи. Третий режим: одна из ЭВМ решает задачи, другая находится в режиме ожидания, готовая подключиться к работе, либо в режиме проведения профилактических работ. При неисправности одной из ЭВМ она переводится в состояние ремонта, а другая в этом случае работает без резервирования.

Аппаратные средства комплексирования ЕС ЭВМ дополняются соответствующими программными средствами. Для первых ВК ЕС ЭВМ была разработана специальная операционная система ОС-К1. В дальнейшем программные средства, обеспечивающие работу средств комплексирования, включались в состав основной операционной системы ОС ЕС. В ОС ЕС предусмотрено соответствующее программное обеспечение для каждого уровня комплексирования. Для обращения одного процессора к другому по интерфейсу прямого управления служит макро команда ПРЯМАЯ ЗАПИСЬ. В связи с последовательной структурой данных, передаваемых через АКК, предусмотрены два метода доступа: последовательный с очередями и базисный последовательный. В каждом методе свой набор макрокоманд, обеспечивающий работу с помощью АКК. Предусмотрена также программная проверка правильности информации, передаваемой через АКК, с помощью контрольной суммы.

Первым двухпроцессорным комплексом в ЕС ЭВМ был комплекс ВК2П45, структура которого (рис. 2.15) является типовой для двухпроцессорных комплексов в ЕС. Тип структурной организации – МПВК с многовходовой оперативной памятью. Процессоры имеют доступ к модулям памяти через адаптеры памяти (АП), которые и осуществляют

необходимую коммутацию. Минимальная емкость ОЗУ в комплексе 2 Мбайт, максимальная – 8 Мбайт (емкость одного модуля 1 Мбайт). Оба процессора могут обращаться в ОЗУ одновременно, конфликты возникают при обращении двух процессоров к одному АП. Для их разрешения предусмотрена схема, организующая очередь запросов. Схема работает таким образом, что при наличии запросов от двух процессоров обращение их к ОЗУ производится попеременно: обращение дважды подряд одного процессора запрещено.

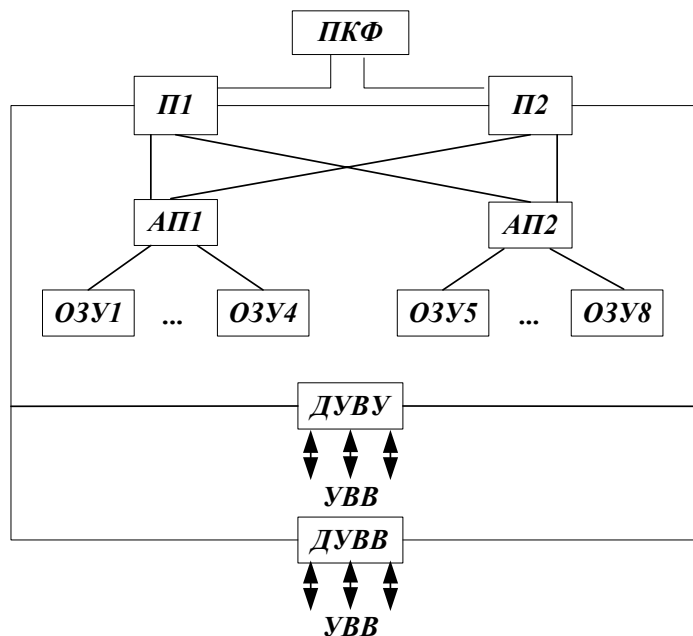


Рис. 2.15 Комплекс ВК2П45

Так как каждый из двух процессоров должен иметь в ОЗУ собственную зону фиксированных ячеек, в качестве которой в однопроцессорном комплексе используется зона с адресами 0–4095, то в каждом процессоре предусмотрен механизм префиксации. Большинство адресов при обращении процессора к памяти обрабатывается с префиксацией. Такие адреса называются реальными, а не обрабатываемые – абсолютными. В результате формирования абсолютного адреса реальные адреса 0–4095 заменяются 4096 адресами блока, адрес которого начинается с адреса, указанного в регистре префикса. Префикс – это 12-разрядное число, размещенное в регистре префикса. Содержимое регистра может быть установлено и проверено командами УСТАНОВИТЬ ПРЕФИКС и ЗАПИСЬ В ПАМЯТЬ ПРЕФИКСА соответственно. При установке префикса разряды 0–7 и 20–31 регистра префикса игнорируются, а при записи в память в эти разряды записываются нули (разряды «обнуляются»). В исходном состоянии во все разряды регистра записаны нули. При префиксации адреса перекодируются следующим образом:

1. Разряды 8–19 адреса памяти, если они имеют нулевые значения, замещаются разрядами 8–19 регистра префикса, т. е. происходит сдвиг зоны с адресами 0–4095 в отведенную данному процессору зону памяти.
2. Разряды 8–19 адреса памяти в случае равенства их разрядам 8–19 префикса заменяются нулями, т. е. часть памяти, отведенная под зону процессора, перемещается в зону с адресами 0–4095.
3. Разряды 8–19 адреса не изменяются, если они все не равны нулю или не равны соответствующим разрядам префикса, т. е. эти адреса являются общими для обоих процессоров и не подлежат изменению.

Реконфигурация комплекса осуществляется со специального пульта реконфигурации, который имеет соответствующие органы. С их помощью любой из модулей ОЗУ можно подключить к любому процессору или к обоим процессорам, а также задать любой диапазон

адресов в пределах установленной емкости ЗУ, соблюдая непрерывность адреса. Кроме того, с пульта осуществляется подключение и реконфигурация УВВ и устанавливается один из трех предусмотренных режимов работы ВК: 1) однопроцессорный, когда комплекс работает как две самостоятельных ЭВМ; 2) двухпроцессорный, при котором функционируют все двухпроцессорные связи; 3) полудуплексный, когда работает один процессор, использующий всю оперативную память, а второй процессор может быть отключен.

В двухпроцессорном, как и в двухмашинном комплексе, существует непосредственная связь между процессорами для обмена управляющей информацией. Потребность в этом обмене может возникнуть при необходимости запуска или остановки одного процессора другим, при взаимном контроле состояния. Кроме упоминавшихся ранее команд ПРЯМАЯ ЗАПИСЬ и ПРЯМОЕ ЧТЕНИЕ в двухпроцессорном режиме для прямого управления используется также специальная команда СИГНАЛ ПРОЦЕССОРУ. Эта команда определяет адрес процессора, с которым осуществляется связь, и код приказа. В адресуемом процессоре могут возникнуть условия, препятствующие выполнению того или иного приказа. Эти условия формируют соответствующие биты банта состояния. Байт состояния передается в выдающий приказ процессора в качестве ответа на поступивший приказ. В двухпроцессорных ВК ЕС предусмотрено 12 таких приказов.

Все указанные аппаратные средства, обеспечивающие построение двухпроцессорных комплексов, поддерживаются специальным вариантом ОС ЕС. Следует отметить, что технические средства ЕС позволяют строить системы не только по типу МПВК с многоходовой памятью. Возможно и построение систем с перекрестной коммутацией, но для этого потребуется разработка коммутатора, которого пока в ЕС ЭВМ не предусмотрен.

ВК на базе СМ ЭВМ. В начале 70-х годов страны социалистического содружества (к ним присоединилась еще и Республика Куба) приступили к совместной разработке Системы малых ЭВМ (мини-ЭВМ). На основе интеграции научно-технического и производственного потенциала в короткий срок была реализована первая очередь малых ЭВМ, включающая в себя четыре ЭВМ (СМ-1 СМ-2, СМ-3 и СМ-4), большой объем периферийного оборудования, а также необходимое программное обеспечение. Система малых ЭВМ была ориентирована на широкое применение: автоматизация технологических процессов, научных исследований, проектирования, использование ЭВМ в качестве интеллектуальных терминалов, решение научных, инженерных и экономических задач.

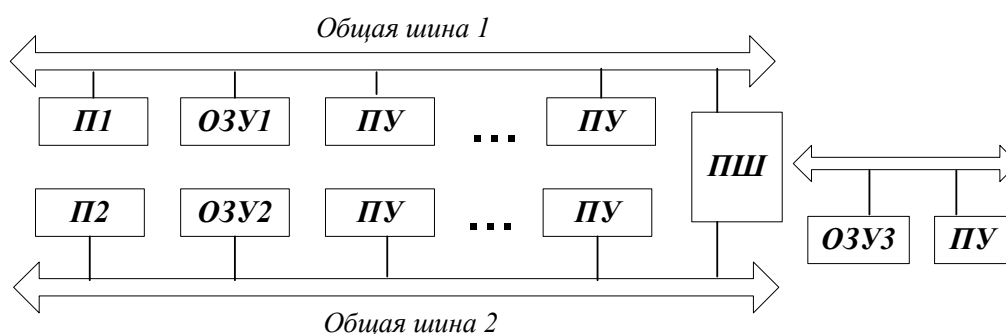


Рис. 2.16. Подключение устройств через переключатель шины

Первая очередь включала в себя четыре базовых процессора СМ-1П, СМ-2П, СМ-3П, СМ-4П производительностью от 200 до 800 тыс. операций в секунду. Первые два продолжили линию отечественных ЭВМ М-6000 и М-7000, получивших широкое распространение в стране, имеющих богатое программное обеспечение и используемых во многих системах управления и автоматизации. Вторые два процессора продолжили линию отечественной ЭВМ М-400, в них реализованы новые структурные и технические решения, характерные для ЭВМ этого класса. Важным моментом, является то, что эти решения в

значительной мере способствуют созданию комплексов и систем на базе этих машин. Основные особенности СМ-3 и СМ-4, созданных на базе этих процессоров, – широкий диапазон производительности в конкретных условиях применения, магистральная (типа «общая шина») структура интерфейса, простая реализация многомашинных и многопроцессорных комплексов, высокая скорость обработки прерываний, гибкость и легкость оснащения устройствами сопряжения с реальными объектами. Все это способствовало тому, что ЭВМ СМ-3 и СМ-4 нашли широкое применение в самых различных областях.

Поскольку ВК в рамках СМ ЭВМ создавались в основном на базе СМ-3 и СМ-4, рассмотрим несколько подробнее структуру и основные особенности этих ЭВМ.

В ЭВМ с одношинной структурой все устройства подключаются к единственной в системе магистрали, называемой общей шиной (см. рис. 2.7, а). Физически общая шина (ОШ) представляет собой магистраль из 56 линий, по которым передается вся информация, необходимая для функционирования ЭВМ как единого комплекса. Такой интерфейс обеспечивает единый способ связи всех устройств, позволяет иметь единый алгоритм связи и унифицированную аппаратуру сопряжения. Все устройства, входящие в состав ЭВМ, используют единый, одинаковый для всех набор сигналов интерфейса.

Как и в ЕС ЭВМ, в СМ ЭВМ предусмотрен набор средств для создания комплексов: переключатель общей шины, адаптер межпроцессорной связи, устройство сопряжения вычислительных машин.

Переключатель общей шины СМ-4501 (ПШ СМ) предназначен для построения на базе процессоров СМ-3П и СМ-4П многомашинных комплексов различной конфигурации.

Переключатель представляет собой электронное устройство, позволяющее подключать к общей шине одного из двух процессоров дополнительный участок шины (ДШ). К ДШ могут подключаться любые устройства СМ ЭВМ, кроме процессора, и в любом наборе (рис. 2.16). Стандарт ДШ полностью соответствует стандарту ОШ, все сигналы ОШ передаются на ДШ без искажений, поэтому процессор, к которому подключены дополнительные устройства, работает с ними, как с собственными. Единственное отличие заключается в том, что ПШ вносит дополнительную задержку при передаче сигнала до 500 нс, что снижает производительность каждой из ЭВМ.

Переключатель общей шины состоит из двух одинаковых по функциональному, электрическому и конструктивному построению секций, каждая из которых подсоединяется к ОШ процессоров. Секция содержит собственно переключатель и расширитель ОШ, обеспечивающий подключение к шине дополнительных устройств.

Таким образом, ПШ СМ позволяет создавать на базе процессора СМ ЭВМ двухмашинные комплексы с общим полем ВЗУ или ОЗУ. При этом суммарная емкость ОЗУ (собственного процессорного и дополнительного) не может превышать максимальной адресуемой (28 кслов для СМ-3П и 128 кслов для СМ-4П). Возможно использование ПШ и просто для резервирования тех или иных устройств с целью повышения надежности. Следует отметить, что дополнительного программного обеспечения при использовании ПШ не требуется.

Адаптер межпроцессорной связи (АМС СМ) предназначен для такой связи между ЭВМ, при которой любой из процессоров двух связываемых ЭВМ может обращаться к устройствам ЭВМ, как к своим собственным. Обращение процессора одной ЭВМ к устройствам другой выполняется обычными командами с использованием так называемого *окна* – зоны адресов, специально отведенной для этой цели. Машина, процессор которой обращается в другую, называется комплексом-источником, а другая ЭВМ – комплексом целью.

Емкость окна выбирается при проектировании комплекса и может составлять 512 кслов, 1, 2, 4, 8, 16, 32 кслов. Местоположение окна на адресной шине комплекса задается программно перед установлением связи через АМС. В качестве окна может быть использован любой незадействованный блок адресов, но на практике чаще всего окно

располагается вслед за адресами оперативной памяти. При работе АМС вносит дополнительную задержку которая не превышает 400 нс. Подключается АМС на ОШ обеих соединяемых с его помощью ЭВМ. Каждое обращение к окну реализуется в режиме прямого доступа. Устройство функционально симметрично, т. е. любая из соединяемых ЭВМ может быть и источником и приемником данных. Обращение через окно может исходить из любого устройства, которое способно быть задающим. Функционально АМС СМ включает в себя несколько адресуемых регистров для входных и выходных данных, адресов и сигналов управления.

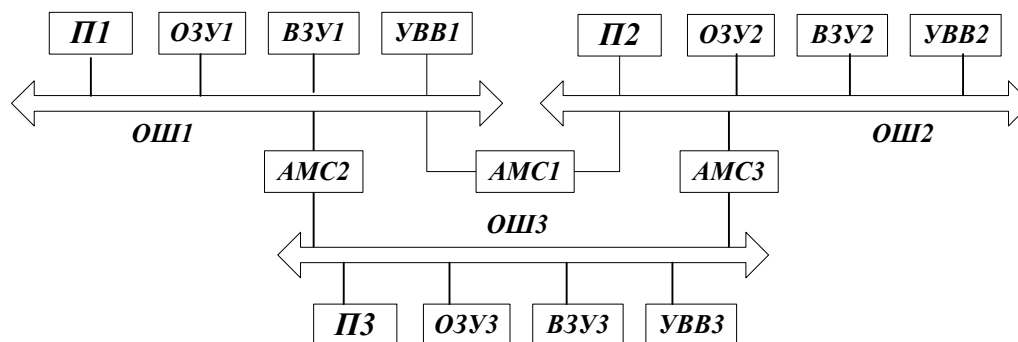


Рис. 2 17. Комплекс на основе АМС

На рис. 2.17 представлен трехпроцессорный симметричный комплекс, в котором операционная система является общей для всех процессоров. Эта же схема может быть и схемой трехмашинного комплекса, если каждый процессор работает только со своей оперативной памятью.

Работа АМС поддерживается программами, входящими в состав специального программного обеспечения СМ ЭВМ.

Для организации иерархических многомашинных комплексов на основе ЭВМ СМ и ЕС в составе технических средств используется устройство сопряжения вычислительных машин (УСВМ) А 71118. Такой ВК включает в себя центральную ЭВМ, функции которой выполняет одна из ЭВМ ЕС и периферийные ЭВМ, в качестве которых используются СМ-3 или СМ-4. Таким образом, УСВМ имеет два интерфейса – интерфейс ЕС ЭВМ и СМ ЭВМ. Со стороны ЭВМ ЕС оно подключается к селекторному или мультиплексному каналу ввода – вывода, со стороны ЭВМ СМ – к ОШ и реагирует на команды ЭВМ как обычное периферийное устройство, используя эти команды для установления связи между ЭВМ и синхронизации их работы.

Устройство сопряжения состоит из двух частей: интерфейсного блока и устройства управления (рис. 2.18). Взаимодействие УСВМ с ЭВМ СМ осуществляется с помощью четырех адресуемых регистров: команд и состояния (РКС), данных (РД), адреса (РА) и длины массива (РДМ). Информация в РКС определяет режим работы УСВМ и отражает его состояние. Реализация алгоритмов взаимодействия ЭВМ ЕС с УСВМ осуществляется аппаратным способом, а инициализация режима обмена – программным. Программы, поддерживающие УСВМ, входят в состав операционной системы «Фобос» СМ.

МПВК «Эльбрус». При создании МПВК «Эльбрус» ставилась задача обеспечить высокую производительность и надежность при решении задач, требующих большого объема вычислений. Наилучшим образом такая комплексная цель может быть достигнута путем создания многопроцессорного комплекса с однородной организацией обработки во всех процессорах. Комплекс «Эльбрус» может содержать до 10 процессоров. При увеличении числа процессоров резко возрастает число конфликтов из-за ресурсов и увеличиваются затраты ресурсов, используемых ОС, в результате чего производительность растет медленно. Для уменьшения этого эффекта применяются следующие меры: увеличивается число модулей оперативной памяти, внешних ЗУ и устройств ввода-вывода,

т. е. число ресурсов; процессоры снабжаются быстродействующими буферными ЗУ достаточно большой емкости; в качестве машинного языка используется алгоритмический язык высокого уровня; наиболее часто выполняемые функции ОС реализуются техническими средствами.

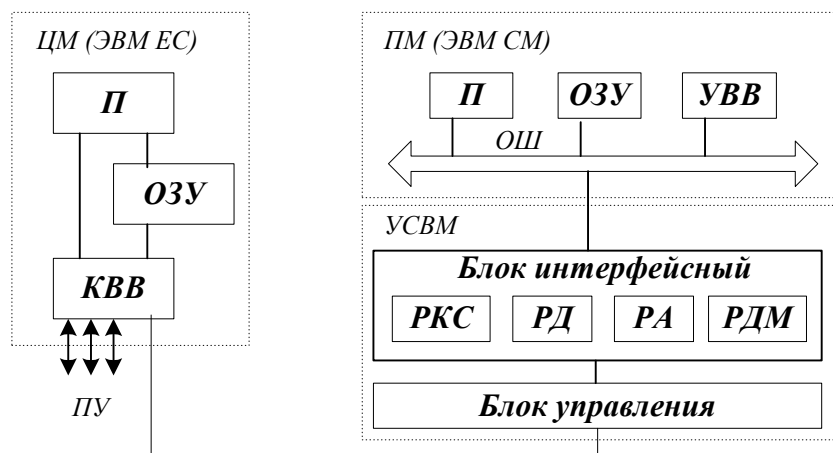


Рис 2.18. Комплекс на основе UCSM

Включение в состав комплекса большого числа модулей с независимым управлением и периферийных устройств позволяет свести к минимуму конфликты из-за ресурсов, необходимых для вычислительных процессов, реализуемых разными процессорами и устройствами. Наличие быстродействующего буферного ЗУ существенно уменьшает число обращений к оперативной памяти, а, следовательно, простои процессоров из-за ожидания памяти. Использование алгоритмического языка высокого уровня уменьшает длину программ (в некоторых случаях в 2–3 раза), в результате чего сокращается число обращений к оперативной памяти. Аппаратная реализация наиболее часто используемых модулей ОС позволяет уменьшить затраты ресурсов на ОС при большом числе процессоров и повысить производительность при малом. Аппаратными средствами реализуются в основном функции диспетчера ОС, что освобождает процессоры от рутинной работы, связанной с организацией очередей, обменом данными, и уменьшает число прерываний, обрабатываемых ОС.

Структурная схема МПВК «Эльбрус» представлена на рис. 2.19. Многоходовая оперативная память построена следующим образом. Четыре модуля оперативной памяти (МП) объединяются одним общим коммутационным модулем (КМ), который подключает любой МП к 10 центральным процессорам (ЦП) и 4 процессорам ввода – вывода (ПВВ). Число КМ в комплексе – до 8, а МП – до 32. Центральные процессоры взаимодействуют с оперативной памятью и ПВВ через КМ.

Все периферийное оборудование (накопители на магнитных лентах, дисках и барабанах, устройства ввода – вывода и передачи данных) подключается к центральной части комплекса через ПВВ, которые являются специализированными, имеют буферное ЗУ, АЛУ и работают по заявкам ЦП. Процессор ввода – вывода реализует аппаратно функции ОС по управлению устройствами ввода – вывода. При этом ПВВ определяет пути обмена информацией (каждое периферийное устройство имеет несколько путей доступа к ПВВ), осуществляет подключение и резервирование устройств. Максимальная скорость обмена ПВВ с ОП достигает 36 Мбайт/с.

В состав внешних ЗУ комплекса входят накопители на магнитных лентах, дисках, барабанах. В качестве НМЛ и НМД используются все типы накопителей, выпускаемых промышленностью в составе ЕС ЭВМ, а НМБ специально разработаны для комплекса «Эльбрус» и имеют следующие характеристики: емкость 4,2 Мбайт, среднее время доступа 5,5 мс, скорость обмена 3,6–3,8 Мбайт/с. Использование НМБ, имеющих существенно

большую скорость, чем НМД, позволяет повысить эффективность обмена между ОЗУ и ВЗУ и в конечном счете увеличить производительность комплекса. При работе с НМБ и НМД используются корректирующие коды, что значительно повышает надежность хранения информации.

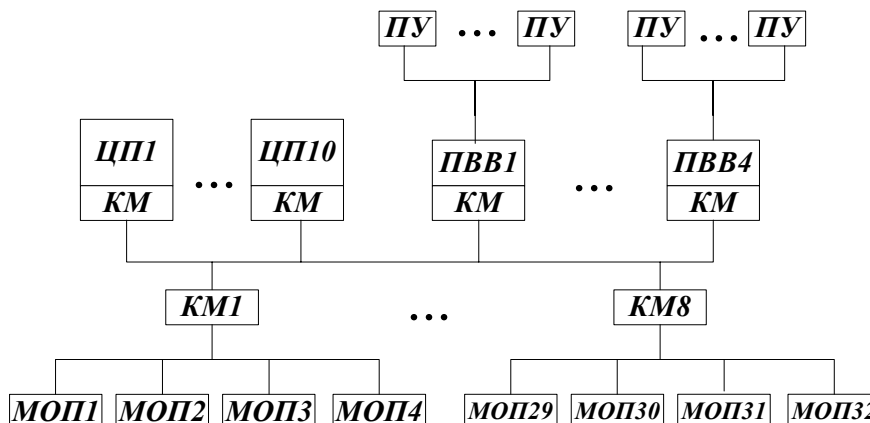


Рис. 2.19. Комплекс «Эльбрус»

В качестве устройств ввода – вывода используется практически весь комплекс устройств ЕС ЭВМ, включая печатающие, перфоленточные, перфокарточные и графические устройства.

Все устройства комплекса работают параллельно и независимо друг от друга и охвачены аппаратным контролем. При возникновении неисправности операционная система получает соответствующий сигнал и осуществляет автоматическую реконфигурацию комплекса, исключая из него неисправное устройство. После восстановления оно может быть вновь введено в рабочий комплект.

Комплекс «Эльбрус» является классическим многопроцессорным не только по структуре, но и по способу организации обработки информации. Все задания и процессы находятся в одной общей очереди, ожидая освобождения процессоров. Как только какой-либо процессор переводит процесс из активного состояния в пассивное, то немедленно обращается к этой очереди и выбирает первый из находящихся в очереди процессов. Последние могут иметь различные приоритеты, тогда фактически организуется несколько очередей процессов.

Комплексы семейства «Эльбрус» отличаются друг от друга производительностью и технической реализацией устройств. Процессоры, применяемые в комплексе «Эльбрус-1», имеют производительность до 1,5 млн. операций в секунду. Время выполнения основных операций: сложение с фиксированной запятой – 520 нс; сложение с плавающей запятой – 780 нс; умножение 32-разрядных чисел – 780 нс; умножение 64-разрядных чисел – 1300 нс; логические операции – 520 нс. Производительность в максимальной комплектации должна достигать 12 млн. операций в секунду.

Комплекс «Эльбрус-2» в максимальной комплектации должен обеспечивать производительность свыше 100 млн. операций в секунду, т. е. примерно в десять раз более высокую, чем «Эльбрус-1». Используемое в «Эльбрус-1» программное обеспечение полностью реализуется в «Эльбрус-2».

2.7. СРАВНЕНИЕ МНОГОМАШИННЫХ И МНОГОПРОЦЕССОРНЫХ КОМПЛЕКСОВ

На практике при необходимости обеспечения высокой производительности и высокой надежности всегда возникает вопрос, каким путем решать эту задачу. Создавать ММВК или МПВК? Среди разработчиков существуют сторонники как многопроцессорных, так и

многомашинных комплексов. Попытаемся оценить преимущества и недостатки того и другого направления.

Наиболее целесообразно оценивать следующие характеристики ВК: надежность, производительность, гибкость, сложность программного обеспечения, экономическую эффективность. Хотя дать точную количественную оценку каждой характеристике невозможно и приведенные ниже оценки носят в значительной степени качественный характер, однако они дают достаточное представление о сравниваемых комплексах.

Все показатели, характеризующие надежность, вне всякого сомнения являются важнейшими, так как зачастую именно для повышения надежности и строятся комплексы. Несложно показать, что по этому параметру МПВК имеет явное преимущество перед ММВК. Это объясняется тем, что в ММВК резервирование осуществляется по машинам, в то время как в МПВК по устройствам. Благодаря этому МПВК может обеспечить более высокую надежность по сравнению с ММВК при том же объеме оборудования или ту же надежность при меньших аппаратных затратах.

Оценка производительности не только любого ВК, но даже однопроцессорной ЭВМ достаточно сложна. Объясняется это прежде всего различием задач по характеру вычислений, т. е. составу операций, по числу обращений к ВЗУ, объему передаваемой между устройствами информации и др. Кроме того, производительность зависит от построения комплекса: структуры, емкости памяти, способов связи между устройствами, параметров самих устройств и др. В силу приведенных обстоятельств нам придется ограничиться качественными оценками производительности ВК.

Рассмотрим два возможных варианта нагрузки комплекса. Первый: в ВК поступает большой поток различных, не связанных между собой задач с небольшим объемом вычислений. Второй вариант: комплекс нагружается крупноразмерными задачами, число которых может быть и невелико, но решение каждой связано с большой вычислительной работой. И ММВК, и МПВК, как уже было показано, характеризуются одинаковой схемой обработки информации: множественный поток команд – множественный поток данных. Из-за разного объема вычислений рабочая нагрузка обрабатывается по-разному.

При первом варианте нагрузки работа ЭВМ в многомашинном комплексе практически мало отличается от режима автономной работы. Общий поток задач тем или иным способом разделяется между всеми ЭВМ, и каждая ЭВМ работает независимо от других. Разделение потока задач может происходить автоматически с использованием общих ВЗУ, где фиксируется список всех задач, в который вносится каждая вновь поступающая в комплекс задача, а ЭВМ по мере окончания решения задач выбирает новые из списка. Существующие связи между ЭВМ используются только с целью обеспечения надежности: если одна из ЭВМ выходит из строя, другая должна продолжать обрабатывать задачи. Объем информации, передаваемой между ЭВМ, незначителен, и, следовательно, малы затраты времени каждой ЭВМ на такой обмен. В результате производительность такого ВК с достаточной точностью может определяться как сумма производительности P_i , всех ЭВМ, входящих в комплекс:

$$P_k = \sum_{i=1}^N P_i$$
 и производительность многомашинного комплекса растет линейно с увеличением числа ЭВМ в нем.

Разумеется, в тех случаях, когда главная задача комплекса – обеспечение высокой надежности и другие ЭВМ работают как резервные (либо решая те же задачи, что и основная, либо находясь в режиме ожидания), производительность комплекса оказывается равной производительности одной ЭВМ, т. е. $P_k = P_i$.

В МПВК также существует общий список задач, который непрерывно пополняется и из которого каждый процессор выбирает новые задачи. На первый взгляд кажется, что организация вычислений в МПВК ничем не отличается от организации в ММВК. Однако это не так, потому что для всех процессоров комплекса общедоступны все ресурсы: оперативная память, периферийные устройства, включая ВЗУ, программные средства; более того, все процессоры работают под управлением единой операционной системы. Общедоступность

всех ресурсов неизбежно приводит к конфликтам между процессорами из-за этих ресурсов, а это означает, что процессоры какую-то часть времени будут неизбежно простаивать. При этом чем больше в комплексе процессоров, тем больше вероятность конфликтов и соответственно простоев, а следовательно, снижения производительности.

Надо учитывать и еще один фактор – затраты времени на организацию процесса обработки информации в ВК, т. е. на работу ОС. Конечно, и в ММВК на работу ОС потребуются определенные затраты машинного времени, и при оценке системной производительности каждой ЭВМ это надо учитывать. Но в МПВК требуются дополнительные затраты на обеспечение взаимодействия всех элементов, прежде всего процессоров. Это связано с тем, что ОС в МПВК, как уже было показано, выполняет более сложные функции, чем в ММВК. Совершенно ясно, что эти затраты будут нелинейно расти с увеличением числа процессоров в системе. Экспериментальные оценки роста производительности МПВК показывают, что при подключении второго процессора производительность увеличивается на 60–80%, но не вдвое. Добавление третьего процессора дает прирост производительности в 2,1 раза, но не втрое.

Другая картина получается при нагрузке комплекса задачами, требующими большого объема вычислений. Обычно эти задачи могут быть расчленены на отдельные части, которые могут решаться параллельно и независимо друг от друга, – как независимые ветви. Но после раздельного решения эти части должны решаться совместно, т. е. подзадачи нельзя рассматривать как совершенно независимые задачи, наоборот, – они представляют собой поток сильносвязанных задач. Организация решения таких крупных задач в ММВК оказывается весьма сложным делом, так как каждая ЭВМ работает под управлением собственной ОС, а требуется совместная работа ЭВМ, т. е. практически надо разрабатывать некоторую надстройку над ОС и эта надстройка должна быть принадлежностью одной из ЭВМ. Последняя становится таким образом, управляющей. Можно попытаться сделать надстройку общей, помещая ее в общее ОЗУ, но в этом случае ММВК превращается практически в МПВК. Второе обстоятельство, приводящее к определенным сложностям при решении в ММВК больших задач, разделенных на части, заключается в том, что объем информации, который необходимо передавать между ЭВМ, оказывается весьма значительным – это исходные данные для подзадач, результаты их решения, промежуточные результаты, информация, необходимая для синхронизации процесса. В конечном счете все это приводит к большим непроизводительным затратам времени всех ЭВМ. Поэтому при решении больших задач производительность комплекса мало увеличивается с ростом числа ЭВМ.

Для МПВК решение таких задач не приводит к заметным дополнительным затратам времени. Точнее, эти затраты на организацию процесса обработки практически мало отличаются от тех, которые имеют место при решении потока независимых задач. Конечно, обеспечение решения сложных задач с параллельной обработкой требует соответствующей ОС, так как организация параллельной обработки и в этом случае далеко не проста.

Таким образом, при решении больших задач производительность МПВК, как правило, превосходит производительность ММВК при равном числе процессоров. Поэтому, когда речь идет о достижении высокой производительности, необходимой для решения больших задач в достаточно короткое время, наиболее широко используются МПВК.

Приведенные рассуждения показывают, что однозначно отдать предпочтение той или иной системе по производительности практически нельзя. Решение может быть принято только с учетом характера решаемых задач.

Такая характеристика ВК, как гибкость, в литературе строго не определена. Будем этим термином определять способность системы к реконфигурации, и в первую очередь к автоматической реконфигурации. Из того, что раньше говорилось о свойствах и особенностях многопроцессорных и многомашинных комплексов, следует, что по этому параметру МПВК существенно превосходит ММВК. Действительно, в ММВК все возможности по созданию новых конфигураций сводятся к отключению и подключению

ЭВМ. В то же время в МПВК возможно создание множества различных конфигураций, причем ОС МПВК рассчитывается на это. Но опять-таки для реализации этого ценного качества в МПВК требуется достаточно сложная система коммутации устройств. К гибкости можно отнести и способность комплекса к наращиванию. В этом отношении преимущество также следует отдать МПВК, для подавляющего большинства которых в силу модульности построения наращивание осуществить достаточно просто.

Сложность программного обеспечения МПВК оценивалась в § 2.4, где было отмечено, что ОС МПВК существенно сложнее ОС ММВК.

Оценка экономической эффективности комплексов может быть дана на основе сравнения характеристик, рассмотренных выше. Она является в известной степени интегральной оценкой, и однозначно определить ее трудно. В самом деле, если оценивать комплексы по надежности, то МПВК имеет явное преимущество, так как эта надежность достигается меньшими затратами. Если говорить о производительности, то МПВК оказывается предпочтительнее только при решении задач, требующих большого объема вычислений. Несомненно преимущество МПВК в отношении эффективного использования технических средств: возможна более высокая загрузка всех без исключения устройств. По сложности операционной системы МПВК заметно уступает ММВК. В связи с тем что в последние годы стоимость аппаратуры заметно снижается, стоимость программного обеспечения приобретает все больший удельный вес в общих затратах на разработку, изготовление и эксплуатацию ВК.

Подводя итог сравнению комплексов с различной организацией, можно констатировать, что однозначного ответа на вопрос о преимуществах того или иного способа организации дать нельзя. В каждом конкретном случае это должно решаться в зависимости от предъявляемых требований по надежности, производительности, от характера задач и рабочей нагрузки.

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

3.1. СИСТЕМЫ С КОНВЕЙЕРНОЙ ОБРАБОТКОЙ ИНФОРМАЦИИ

Принцип конвейерной обработки информации нашел широкое применение в вычислительной технике. В первую очередь это относится к конвейеру команд. Практически все современные ЭВМ используют этот принцип. Вместе с тем во многих вычислительных системах наряду с конвейером команд используется и конвейер данных. Сочетание этих двух конвейеров дает возможность достигнуть очень высокой производительности систем на определенных классах задач, особенно если при этом используется несколько конвейерных процессоров, способных работать одновременно и независимо друг от друга. Именно так и построены самые высокопроизводительные системы. Целесообразнее всего рассмотреть принцип конвейерной обработки на примере некоторых, наиболее представительных систем.

К числу ЭВМ, в которых широкое применение нашел конвейер команд, относится одна из лучших отечественных машин БЭСМ-6. Эта ЭВМ, разработанная под руководством академика С. А. Лебедева в 1966 г., была в течение многих лет самой быстродействующей в стране благодаря целому ряду интересных решений, в том числе и конвейеру команд. Последний обеспечивался использованием восьми независимых модулей ОЗУ, работающих в системе чередования адресов, и большого числа быстрых регистров, предназначенных также и для буферизации командной информации. Это позволило получить на БЭСМ-6 производительность 1 млн. операций в секунду.

Определенный интерес представляет построение систем IBM 360/91, а также более поздней и более современной IBM 360/195. Пять основных устройств системы: ОЗУ, управления памятью с буферным ОЗУ, процессор команд, операционные устройства для выполнения операций с плавающей запятой, с фиксированной запятой и десятичной арифметики работают одновременно и независимо друг от друга. Оперативное ЗУ построено по многомодульному принципу (до 32 модулей), устройство управления памятью работает по принципу конвейера и обеспечивает 8- или 16-кратное чередование адресов при обращении процессора и каналов ввода – вывода информации к ОЗУ.

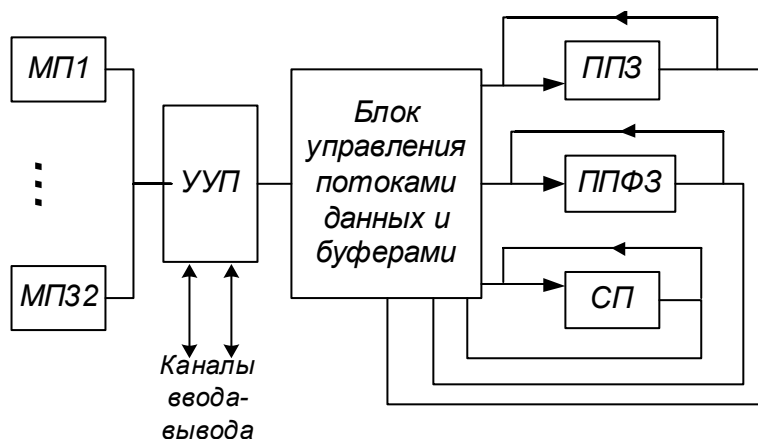


Рис. 3.1. Система STAR-100

Кроме конвейера команд в системах IBM 360/91 и IBM 370/195 в обоих ОУ используется также и конвейерная обработка данных. Однако в системе 360/195 конвейер получается довольно внушительный: в каждом цикле осуществляется выборка до 8 команд,

расшифровка 16 команд, до 3 операций над адресами и до 3 процессорных операций. Всего в системе одновременно может обрабатываться до 50 команд.

Существенно более полно используется принцип магистральной обработки в системе STAR-100, разработанной фирмой CDC в 1973 г. Надо сказать, что фирма CDC в большинстве разрабатываемых и выпускаемых машин и систем использует принцип конвейерной обработки для повышения производительности, однако раньше использовался только командный конвейер, а в системе STAR-100 – оба типа конвейера. Система содержит три конвейерных процессора (рис. 3.1): ППЗ – процессор, содержащий конвейерные устройства сложения и умножения с плавающей запятой; ППФЗ – процессор, содержащий конвейерное устройство сложения с плавающей запятой, конвейерное многоцелевое устройство, выполняющее умножение с фиксированной запятой, деление и извлечение квадратного корня; СП – специальный конвейерный 16-разрядный процессор, выполняющий операции с фиксированной запятой и ряд логических операций.

Конвейерные процессоры оперируют с 64- или 32-разрядными числами и каждые 40 нс выдают результаты в блок управления потоками данных и буферами. Оперативное ЗУ построено по модульному принципу (32-модуля памяти) и работает с чередованием адресов под управлением устройства УП (управление памятью). Каждый малый цикл обращения к памяти – 40 нс. (Полный цикл ОЗУ–1,28 мкс, т. е. 40X32 нс.)

Конвейерный сумматор с плавающей запятой состоит из четырех сегментов – специализированных операционных устройств (см. рис. 2.2, а). Продолжительность цикла каждого сегмента составляет 40 нс; таким образом, время выполнения операции сложения с плавающей запятой равно 160 нс.

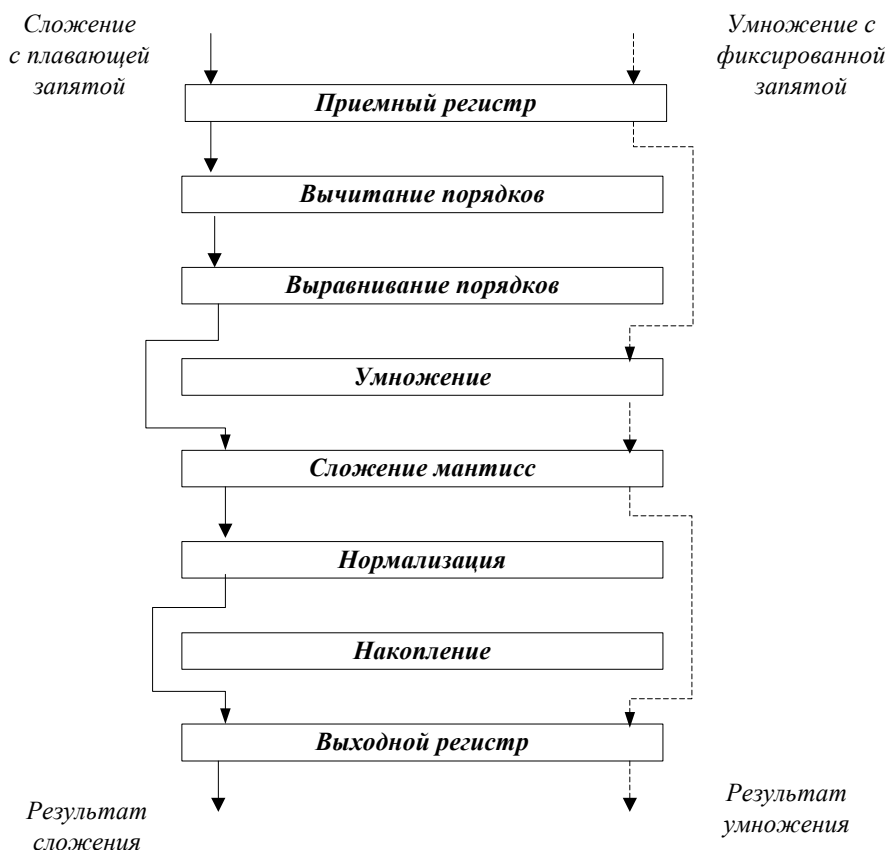


Рис. 3.2. Конвейер системы ASC

Конвейерный умножитель включает в себя 8 сегментов, поэтому время выполнения операции умножения составляет 320 нс. Но при загрузке конвейерных процессоров длинной последовательностью операндов, над которыми производится одна и та же операция,

результат выдается каждые 40 нс. Учитывая, что каждый из двух основных процессоров может выдавать по два 32-разрядных результата, нетрудно подсчитать, что система STAR-100 может в пределе выполнять до 100 млн. операций в секунду.

Устройства конвейерной обработки далеко не всегда выполняют с жесткой настройкой на одну определенную операцию. Чаще их делают многоцелевыми, вводя в конвейер сегменты, необходимые для реализации полного набора операций, в процессе выполнения которых весь тракт настраивается соответствующим образом. На рис. 3.2 представлена структура системы ASC фирмы «Техас Инструменте» и показано, какие сегменты универсальной цепочки работают при различных операциях.

Одной из наиболее высокопроизводительных вычислительных систем в мире общепризнанно считается система СКАУ, созданная в 1976 г. В этой системе конвейерный принцип обработки используется в максимальной степени: имеется и конвейер команд, и конвейер арифметических и логических операций. Кроме того, в системе широко применяется совмещенная обработка информации несколькими устройствами. Все это позволило при решении научных задач достигнуть чрезвычайно высокой производительности – до 250 млн. операций в секунду

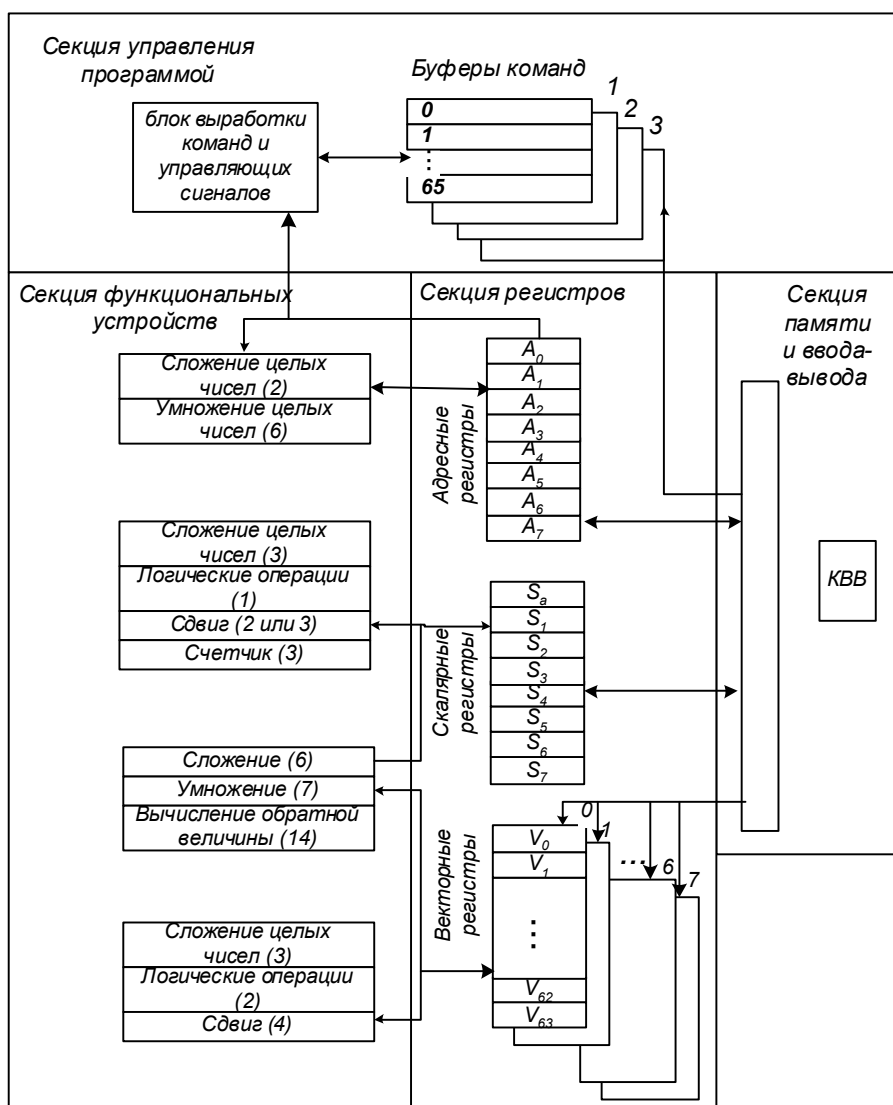


Рис. 3.3. Система CRAY

Система CRAY (рис. 3.3) состоит из четырех секций: функциональных устройств, регистров, управления программой, памяти и ввода – вывода. В системе 12 функциональных устройств, работающих в режиме конвейера, разбитых на 4 группы: адресную, скалярную,

операций с плавающей запятой и векторную. Число сегментов в каждом функциональном устройстве (указано в скобках на схеме) сравнительно невелико, оно зависит от сложности операций и колеблется в пределах от 1 до 14 (вычисление обратной величины). Такое сравнительно небольшое число сегментов в каждом магистральном устройстве имеет определенные преимущества – они сравнительно быстро заполняются. Длительность цикла каждого сегмента составляет 12,5 нс: это значит, что каждые 12,5 нс любое функциональное устройство может выдавать результаты.

Оперативная память системы, выполненная на интегральных схемах, имеет емкость 1 млн. слов (позже была увеличена до 4 млн.) и организована в виде 16 блоков памяти с независимым управлением емкостью по 64 кслов. Каждый блок включает в себя 72 модуля, причем модуль содержит один разряд всех 64 кслов. Система работает с 64-разрядными словами, 8 разрядов используется для коррекции одиночных и обнаружения двойных ошибок, что обеспечивает высокую надежность хранения информации. Независимые блоки дают возможность организовать 16-кратное чередование адресов. Цикл обращения к памяти – 50 нс.

Существенную роль в достижении столь высокой производительности играют быстрые регистры. Они разделены на 3 группы: адресные – А-регистры, скалярные – S-регистры и векторные – V-регистры. Адресные регистры 24-разрядные, их всего восемь; 64-разрядных 5-регистров также восемь и восемь 64-элементных V-регистров, причем каждый элемент вектора содержит 64-разрядное слово. Время обращения к регистру всего лишь 6 нс. В системе имеется еще две группы промежуточных регистров (между ОЗУ и А-, S- и V-регистрами): 24-разрядные B-регистры и 64-разрядные T-регистры, на рисунке не показанные. Все эти регистры позволяют конвейерным устройствам работать с максимальной скоростью без непосредственного обращения к ОЗУ: все операнды получают из регистров и результаты отправляются также в регистры. Благодаря регистрам конвейерные устройства связываются в цепочки, т. е. поток результатов засылаемых в векторный регистр одним устройством, одновременно служит входным потоком операндов для другого устройства; исключаются промежуточные обращения к памяти. Это является еще одной отличительной особенностью системы CRAY, повышающей ее производительность.

Состав операций универсальный, только вместо деления используется операция вычисления обратной величины. Общее число операций 128. Команды двух форматов – 16 и 32 разряда. Арифметические и логические команды имеют 16-разрядный формат 7 разрядов – код операции и по 3 разряда для адресов регистров операндов и результата, причем 6 разрядов адресов регистров операндов в совокупности с дополнительными 16 разрядами используются для обращения к основной памяти и командам перехода.

Ввод – вывод информации осуществляется через 24 канала, сгруппированных в 4 группы, причем в каждой группе имеются либо каналы ввода, либо каналы вывода информации. Обмен осуществляется двухбайтными кодами. Для связи с внешними абонентами используется периферийная ЭВМ.

Высокая производительность системы CRAY обеспечивается и другими факторами.

1. Конструкция ЭВМ весьма компактна, благодаря чему время передачи сигналов между устройствами мало, и это позволяет работать с тактом 12.5 нс.

2. Используется гибкая система адресации выборка из массивов может осуществляться по строкам, столбцам и диагоналям с произвольным постоянным шагом.

3. В состав системы входит подсистема дисковой памяти из четырех контроллеров, каждый из которых управляет четырьмя накопителями общей емкостью 76 854 млрд. бит.

4. Система имеет достаточно современное программное обеспечение, в том числе: операционную систему, рассчитанную на пакетную мультипрограммную обработку 63 задач; оптимизирующий компилятор с фортрана, автоматически распознающий циклы, удобные для реализации векторными командами; макроассемблер, библиотеку стандартных программ, загрузчик и другие средства.

Все это в совокупности и дает основание считать системы CRAY наиболее высокопроизводительными.

3.2. МАТРИЧНЫЕ СИСТЕМЫ

Наиболее распространенными из систем класса ОКМД являются матричные системы, которые лучше всего приспособлены для решения задач, характеризующихся параллелизмом независимых объектов или параллелизмом данных. Организация систем этого типа на первый взгляд достаточно проста: общее управляющее устройство, генерирующее поток команд, и большое число устройств, работающих параллельно и обрабатывающих каждое свой поток данных. Таким образом, производительность системы оказывается равной сумме производительности всех обрабатывающих устройств. Однако на практике, чтобы обеспечить достаточную эффективность системы при решении широкого круга задач, необходимо организовать связи между обрабатывающими устройствами (в дальнейшем будем называть их процессорными элементами – ПЭ). Характер связей может быть различным, так же как и характер взаимодействия ПЭ. Все это и определяет разные свойства систем.

Если не считать машины Унгера (1958 г.), которая была узкоспециализированной и предназначалась только для решения задач распознавания образов, то, по-видимому, первой матричной системой следует считать систему SOLOMON упрощенная структура которой представлена на рис. 3.4. Система содержит 1024 ПЭ, соединенных в виде матрицы 32X32. Каждый ПЭ в матрице соединен с четырьмя соседними и включает в себя процессор, обеспечивающий выполнение последовательных поразрядных арифметических и логических операций, а также оперативное ЗУ емкостью 16 Кбит, разбитое на модули по 4 Кбит каждый. Длина слова переменная – от 1 до 128 разрядов. Разрядность слов устанавливается программно. По каналам связи от УУ передаются команды и общие константы. В ПЭ используется так называемая многомодальная логика, которая позволяет каждому ПЭ выполнять (т. е. быть активным) или не выполнять (быть пассивным) общую операцию в зависимости от значений обрабатываемых данных.

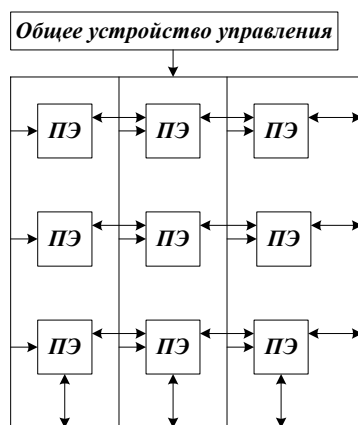


Рис. 3.4. Система типа ОКМД

В каждый момент все активные ПЭ выполняют одну и ту же операцию над данными, хранящимися в собственной памяти и имеющими один и тот же адрес. Идея многомодальности заключается в том, что в каждом ПЭ имеется специальный регистр на четыре состояния – *регистр моды*. Мода (или модальность) заносится в этот регистр от УУ. При выполнении последовательности команд модальность передается в коде операции и сравнивается с содержимым регистра моды. Если есть совпадение, то операция выполняется. В других случаях ПЭ не выполняет операцию, но может в зависимости от кода пересылать свои операнды соседнему ПЭ. Такой механизм позволяет, в частности, выделить строку или столбец ПЭ, что может быть полезным при операциях над матрицами. Взаимодействуют ПЭ с периферийным оборудованием через внешние ПЭ.

Система SOLOMON оказалась нежизнеспособной вследствие громоздкости, недостаточной гибкости и эффективности. Однако идеи, заложенные в ней, получили развитие в системе ILLIAC-IV, разработанной Иллинойским университетом и изготовленной фирмой «Барроуз».

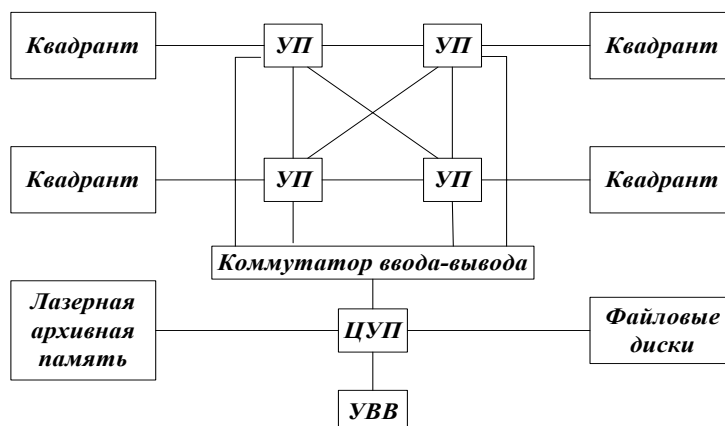


Рис. 3.5. Система ILLIAC-IV (проект)

По первоначальному проекту система ILLIAC-IV должна была включать в себя 256 ПЭ, разбитых на 4 группы – квадранты (рис 35), каждый из которых должен управляться специальным процессором (УП). Управление всей системой, содержащей кроме ПЭ и УП также внешнюю память и оборудование ввода – вывода, предполагалось от центрального управляющего процессора (ЦУП). Однако реализовать этот замысел не удалось из-за возникших технологических трудностей при создании интегральных схем, ОЗУ и удорожания всего проекта почти в два раза. В результате с опозданием на два года (в 1971 г.) система была создана в составе одного квадранта и одного УП и с начала 1974 г. введена в эксплуатацию. Ранее предполагалось получить на этой системе производительность примерно 1 млрд. операций в секунду, однако реализовано 200 млн. операций в секунду. Тем не менее этого оказалось достаточно, чтобы система в течение ряда лет считалась самой высокопроизводительной в мире.

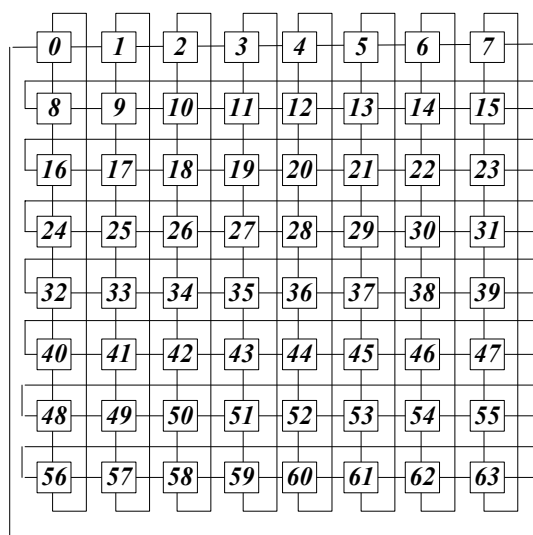


Рис. 3.6. Квадрант системы ILLIAC-IV

В каждом квадранте 64 ПЭ образуют матрицу размером 8x8. Схема связей между ПЭ (рис. 3.6) похожа на схему системы SOLOMON, но связь с внешней средой имеют все ПЭ без исключения. Реально действующая система ILLIAC-IV состоит, таким образом, из двух частей: центральной с устройством управления и 64 ПЭ, а также подсистемы ввода – вывода, включающей в себя универсальную ЭВМ В-6700, файловые диски и лазерную архивную память большой емкости. Каждый ПЭ состоит из собственно процессора и ОЗУ. Процессор оперирует с 64-разрядными

числами и выполняет универсальный набор операций. Быстродействие процессора достаточно высокое, операция сложения 64-разрядных чисел выполняется за 240 нс, а умножения – за 400 нс. Таким образом, процессор выполняет в среднем 3 млн. операций в секунду, а следовательно производительность системы равна $3 \times 64 \approx 200$ млн. операций в секунду.

Емкость ОЗУ каждого ПЭ составляет 2048 64-разрядных слов, длительность цикла обращения к памяти 350 нс. Память выполнена на интегральных схемах. Каждый процессор имеет счетчик адресов и индексный регистр, так что конечный адрес в каждом процессоре может формироваться как сумма трех составляющих: адреса, содержащегося в команде для данного ПЭ, кода, содержащегося в центральном индексном регистре УУ, и кода, содержащегося в собственном индексном регистре. Это существенно повышает гибкость системы по сравнению с системой SOLOMON, где все ПЭ выбирают информацию по одному адресу. Каждый процессор кроме индексного регистра имеет в своем составе пять программно-адресуемых регистров: накапливающийся сумматор, регистр для операндов, регистр пересылок, используемый при передачах от одного ПЭ к другому, буферный регистр на одно слово и регистр управления состоянием ПЭ (аналогичный регистру моды в системе SOLOMON). Регистр управления имеет 8 разрядов. В зависимости от содержимого этого регистра ПЭ становится активным или пассивным, а также выполняет ряд пересылочных операций. Если вычисления не требуют полной разрядности, то процессор может быть разбит на два 32-разрядных подпроцессора или даже восемь 8-разрядных. Это позволяет в случае необходимости обрабатывать векторные операнды из 64, $2 \times 64 = 128$ и $8 \times 64 = 512$ элементов.

Как видно из рис. 3.6, каждый i -й ПЭ связан с четырьмя другими: $(i-1)$ -, $(i+1)$ -, $(i+8)$ - и $(i-8)$ -м. При такой связи передача данных между любыми двумя ПЭ осуществляется не более, чем за 7 шагов, а среднее число шагов равно 4.

По шине состояния ПЭ могут передавать сигналы о состоянии в УУ, которое таким образом всегда определяет состояние системы. Оперативное ЗУ каждого процессорного элемента связано со своим процессором, устройством управления центральной частью и подсистемой ввода – вывода.

Подсистема ввода – вывода включает в себя стандартную ЭВМ В-6700 (первоначально В-6500) и два уровня внешней памяти: на магнитных дисках с фиксированными дорожками и лазерную память. Накопители на дисках имеют магнитные головки для каждой дорожки (128 головами на диск), и обмен данными осуществляется по 256-разрядной шине. Емкость каждого диска – около 1 млрд. бит. Для того чтобы согласовать скорость передачи информации с дисков и работу управляющей ЭВМ, в систему включено буферное ОЗУ, состоящее из четырех модулей памяти.

Лазерная память представляет собой одностороннее ЗУ очень большой емкости (10^{12} бит). Информация записывается на тонкой металлической пленке путем прожигания микроотверстий лазерным лучом. Емкость ЗУ – 1200 млрд. бит. Время доступа к данным от 0,2 до 5 с.

Машина В-6700 выполняет и различные другие функции: транслирует и компонирует программы, управляет запросами на ресурсы, производит предварительную обработку информации и т. д.

Следует подчеркнуть, что сверхвысокая производительность системы достигается только на определенных типах задач, таких, например, как операции над матрицами, быстрое преобразование Фурье, линейное программирование обработка сигналов, где как раз имеет место параллелизм данных или параллелизм независимых объектов. Необходимо отметить также и то, что разработка программ для систем ILLIAC-IV, обеспечивающих высокую производительность, является весьма сложным делом. Для упрощения этой работы были разработаны специальные алгоритмические языки.

Система ILLIAC-IV была включена в состав вычислительной сети ARPA. В результате совершенствования программного обеспечения производительность системы выросла до 300 млн. операций в секунду.

В начале 80-х годов в Советском Союзе в классе систем ОКМД была создана «Параллельная система 2000» (ПС-2000), которая также может быть отнесена к матричным [5]. Эта система ориентирована на решение задач, характеризующихся параллелизмом данных, независимых ветвей и объектов. Разработчики системы считают, в частности, что ПС-2000 найдет особенно широкое и эффективное применение для обработки геофизической информации, получаемой при поиске нефти и газа, решения задач плазменной кинетики,

расчетов устойчивости летательных аппаратов, обработки гидролокационных сигналов и изображений, решения задач в частных производных и др.

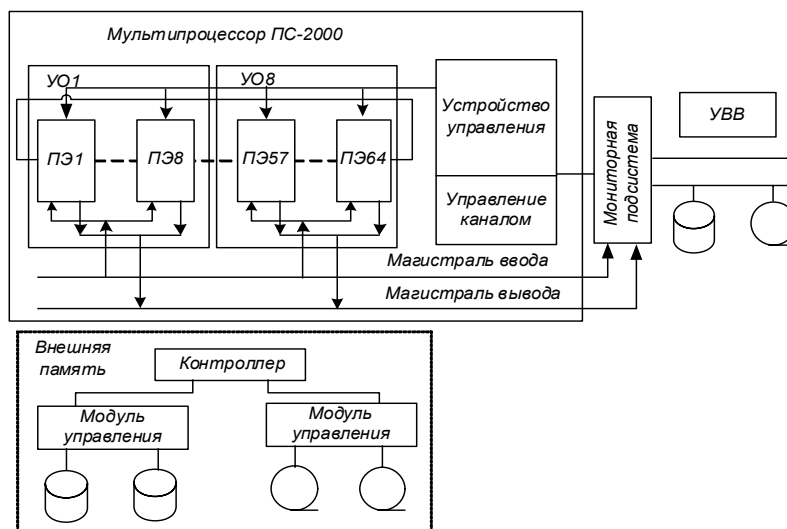


Рис. 3.7. Система ПС-2000

Укрупненная структура системы ПС-2000 изображена на рис. 3.7. Центральная часть системы – мультипроцессор ПС-2000, состоящий из решающего поля и устройства управления мультипроцессором. Решающее поле строится из одного, двух, четырех или восьми устройств обработки (УО), в каждом из которых восемь процессорных элементов. Таким образом, мультипроцессор может содержать 8, 16, 32 или 64 процессорных элемента. Процессорный элемент обрабатывает 24-разрядные слова, используемые для представления 12-, 16- и 24- разрядных чисел с фиксированной запятой, в которых 20 разрядов определяют мантиссу и 4 разряда – шестнадцатеричный порядок. Емкость оперативной памяти процессорного элемента – 4096 или 16 384 24-разрядных слова с циклом обращения соответственно 0,64 или 0,94 мкс. Мультипроцессор из 64 процессорных элементов обеспечивает при обработке данных следующую производительность (в миллионах операций в секунду):

Сложение с фиксированной запятой (регистр–регистр)	200
Сложение с плавающей запятой	66,4
Умножение	28,5 – 50,0

Время выполнения основных операций (в микросекундах):

Транспонирование матрицы 64X64	0,45
Умножение матриц 64X64:	
с фиксированной запятой	1,0
с плавающей запятой	1,4
Быстрое преобразование Фурье на 1024 комплексные точки:	
с фиксированной запятой	1,0 – 2,5
с плавающей запятой	1,4 – 2,8

Устройство управления содержит блок микропрограммного управления емкостью 16384 микрокоманды, ОЗУ емкостью 4096 или 16384 24-разрядных слов и АЛУ. В блок микропрограммного управления загружаются микропрограммы обработки данных, ввода – вывода и управления. В комплект программного обеспечения системы входит набор микропрограмм базовой системы операций, ориентированной, на обработку матриц, реализацию быстрого преобразования Фурье и решение задач математической статистики, спектрального анализа, линейного и динамического программирования. Возможна

реализация на микропрограммном уровне любой необходимой системы операции. В ОЗУ загружается программа обработки. Команды программы выполняются в обычном порядке. Выполнение сводится к инициированию соответствующей микропрограммы, микрокоманды которой формируют управляющие сигналы, воздействующие на процессорные элементы.

Ввод – вывод данных в память решающего поля, т. е. в ОЗУ процессорных элементов, производится через канал прямого доступа, состоящий из магистралей ввода и вывода. Операции ввода – вывода инициируются мониторной подсистемой и выполняются под управлением устройства управления мультипроцессора. Обработка, ввод и вывод данных могут выполняться одновременно. Обмен данными может осуществляться до 1,8 Мбайт/с при вводе и до 1,4 Мбайт/с при выводе. Процессорные элементы решающего поля связаны регулярным каналом, по которому передаются данные. Конфигурация регулярного канала перестраивается под управлением программы, образуя одно кольцо из 64 процессорных элементов, как на рисунке, или несколько одинаковых колец: 8 колец по 8 процессорных элементов, 4 по 16 или 2 по 32.

В целом система управляется мониторной подсистемой, состоящей из одной или двух мини-ЭВМ СМ-2 и подключенных к ним периферийных устройств. На ЭВМ реализуется подготовка микропрограмм и программ для мультипроцессора и программ для самой мониторной подсистемы с использованием языков программирования и соответствующих трансляторов. Мониторная подсистема управляет загрузкой микропрограмм в мультипроцессор, контролирует работу мультипроцессора и обеспечивает обмен данными между системой и пользователями.

Для хранения больших объемов данных система ПС-2000 снабжена внешней памятью, состоящей из микропрограммируемого контроллера, четырех накопителей на сменных магнитных дисках и восьми накопителей на магнитных лентах. Накопители подключаются к контроллеру через модули управления (локальные контроллеры).

Вычислительный процесс в системе ПС-2000 складывается из трех составляющих: процесса в мониторной подсистеме, выполнения программы, ввода – вывода данных. Процесс в мониторной подсистеме является основным: он инициирует остальные процессы и синхронизирует их. Под управлением мониторной подсистемы в устройство управления загружается набор микропрограмм и программа. В процессе обработки данных набор микропрограмм и программа могут изменяться. Этапы обработки перемежаются с этапами обмена данными между решающим колом мультипроцессора, его внешней памятью и мониторной подсистемой.

Контроль работоспособности системы обеспечивается схемными средствами, контролирующими корректность хранения и передачи данных, и набором программ для проверки функционирования системы. Для поиска неисправностей используется система диагностических программ.

Программное обеспечение ППС-2000 разработано на базе агрегатной системы программного обеспечения (АСПО) СМ ЭВМ, к которому добавлены модули, организующие работу ППС-2000.

Система ПС-2000 имеет ряд преимуществ перед ранее разработанными матричными системами, в частности системой ILLIAC-IV.

1. Процессорные элементы ППС-2000 имеют существенно большие возможности, что определяется наличием сверхоперативной регистровой памяти, отдельного процессора для операций над адресами, процессора активации, функции которого значительно шире, чем функции регистра моды в системе ILLIAC-IV.

2. Благодаря наличию собственной памяти и индексной арифметики, используемой для организации счетчиков адресов, возможно совмещение обмена информацией между модулями памяти ПЭ, АЛУ и устройствами ввода – вывода.

3. Достаточно большая емкость памяти в каждом ПЭ (16 кслов) в сочетании с многоуровневой системой прерываний делает возможной организацию мультипрограммного режима с выделением независимых ресурсов для каждой задачи.

4. Двухуровневое управление (командное и микрокомандное) обеспечивает более эффективное программирование.

5. Возможно наращивание системы модулями по восемь процессорных элементов, причем и конструкция, и набор команд позволяет это делать без изменения средств управления.

6. Стоимость системы сравнительно невысока.

Опыт использования первых систем ПС-2000 показал, что при решении типичных задач геофизики, ядерной физики, аэродинамики и других система обеспечивает высокую системную производительность – до 200 млн. операций сложения с фиксированной запятой в секунду.

3.3. АССОЦИАТИВНЫЕ СИСТЕМЫ I

К числу систем класса ОКМД относятся ассоциативные системы. Эти системы, как и матричные, характеризуются большим числом операционных устройств, способных одновременно, по командам одного управляющего устройства вести обработку нескольких потоков данных. Но эти системы существенно отличаются от матричных способами формирования потоков данных. В матричных системах данные поступают на обработку от общих или отдельных запоминающих устройств с адресной выработкой информации либо непосредственно от устройств – источников данных. В ассоциативных системах информация на обработку поступает от ассоциативных запоминающих устройств (АЗУ), характеризующихся тем, что информация из них выбирается не по определенному адресу, а по ее содержанию.

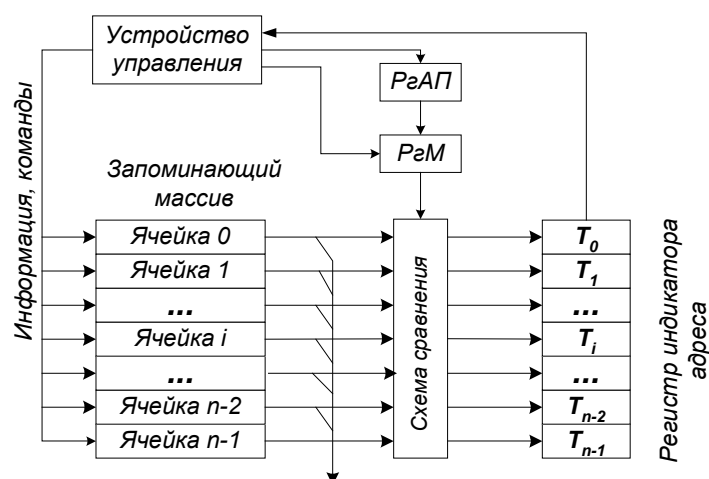


Рис 38 Ассоциативное запоминающее устройство

Принцип работы АЗУ поясняет схема, представленная на рис. 3.8. Запоминающий массив, как и в адресных ЗУ, разделен на m -разрядные ячейки, число которых n . Практически для любого типа АЗУ характерно наличие следующих элементов: запоминающего массива; регистра ассоциативных признаков (РгАП); регистра маски (РгМ); регистра индикаторов адреса со схемами сравнения на входе. В АЗУ могут быть и другие элементы, наличие и функции которых определяются способом использования АЗУ.

Выборка информации из АЗУ происходит следующим образом. В РгАП из устройства управления передается код признака искомой информации (иногда его называют компарандом). Код может иметь произвольное число разрядов – от 1 до m . Если код признаков используется полностью, то он без изменения поступает на схему сравнения, если же необходимо использовать только часть кода, тогда ненужные разряды маскируются с помощью РгМ. Перед началом поиска информации в АЗУ все разряды регистра индикаторов адреса устанавливаются в состояние 1. После этого производится опрос первого разряда всех

ячеек ЗМ и содержимое сравнивается с первым разрядом РГАП. Если содержимое первого разряда i -й ячейки не совпадает с содержимым первого разряда РГАП, то соответствующий этой ячейке разряд регистра индикаторов адреса T_i сбрасывается в состояние 0, если совпадает, – на T_i остается 1. Затем эта операция повторяется со вторым, третьим и последующими разрядами до тех пор, пока не будет произведено сравнение со всеми разрядами РГАП. После поразрядного опроса и сравнения в состоянии 1 останутся те разряды регистра индикаторов адреса, которые соответствуют ячейкам, содержащим информацию, совпадающую с записанной в РГАП. Эта информация может быть считана в той последовательности, которая определяется устройством управления.

Заметим, что время поиска информации в ЗМ по ассоциативному признаку зависит только от числа разрядов признака и от скорости опроса разрядов, но совершенно не зависит от числа ячеек ЗМ. Этим и определяется главное преимущество АЗУ перед адресными ЗУ: в адресных ЗУ при операции поиска необходим перебор всех ячеек запоминающего массива.

Запись новой информации в ЗМ производится без указания номера ячейки. Обычно один из разрядов каждой ячейки используется для указания ее занятости, т. е. если ячейка свободна для записи, то в этом разряде записан 0, а если занята, – 1. Тогда при записи в АЗУ новой информации устанавливается признак 0 в соответствующем разряде РГАП и определяются все ячейки ЗМ, которые свободны для записи. В одну из них устройство управления помещает новую информацию.

Нередко АЗУ строятся таким образом, что кроме ассоциативной допускается и прямая адресация данных, что представляет определенные удобства при работе с периферийными устройствами.

Необходимо отметить, что запоминающие элементы АЗУ в отличие от элементов адресуемых ЗУ должны не только хранить информацию, но и выполнять определенные логические функции, поэтому позволяют осуществить поиск не только по равенству содержимого ячейки заданному признаку, но и по другим условиям: содержимое ячейки больше (меньше) признака РГАП, а также больше или равно (меньше или равно).

Отмеченные выше свойства АЗУ характеризуют преимущества АЗУ для обработки информации. Формирование нескольких потоков идентичной информации с помощью АЗУ осуществляется быстро и просто, а с большим числом операционных элементов можно создавать высокопроизводительные системы. Надо учитывать еще и то, что на основе ассоциативной памяти легко реализуется изменение места и порядка расположения информации. Благодаря этому АЗУ является эффективным средством формирования наборов данных.

Исследования показывают, что целый ряд задач, таких, как обработка радиолокационной информации, распознавание образов, обработка различных снимков и других задач с матричной структурой данных, эффективно решается ассоциативными системами. К тому же программирование таких задач для ассоциативных систем гораздо проще, чем для традиционных.

Наиболее характерным представителем группы ассоциативных вычислительных систем является система STARAN, разработанная в США. От матричных систем, описанных выше, она отличается не только наличием ассоциативной памяти, но и другими особенностями, ассоциативная память является памятью с многомерным доступом, т. е. в нее можно обратиться как поразрядно, так и пословно, операционные процессорные элементы предусмотрены для каждого слова памяти; имеется уникальная схема перестановок для перегруппировки данных в памяти.

Основным элементом системы является многомерная ассоциативная матрица – ассоциативный модуль (АМ), который представляет собой квадрат из 256 разрядов на 256 слов, т. е. содержит в общей сложности 65536 бит данных. Для обработки информации имеется 256 процессорных элементов, которые последовательно, разряд за разрядом, обрабатывают слова (рис. 3.9). Все ПЭ работают одновременно, по одной команде, выдаваемой устройством управления. Таким образом, сразу по одной команде обрабатываются все выбранные по определенным признакам из памяти слова.

Схема перестановок позволяет сдвигать и перегруппировывать данные так, чтобы над словами, хранящимися в памяти, можно было выполнять параллельно арифметические и логические операции. Большая часть операций выполняется в отношении каждого из 256-разрядных слов. Операции, в которых участвуют несколько слов, используются достаточно редко. Обычно 250-разрядное слово ассоциативной матрицы разбивается программистом на поля переменной длины, и в процессе обработки именно над этими полями производятся и арифметические и логические действия.

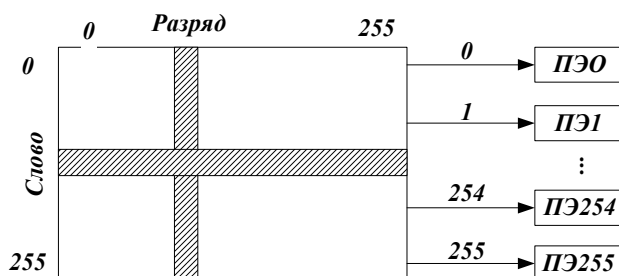


Рис. 3.9. Процессорная обработка в системе STARAN

Базовая конфигурация системы STARAN содержит один АМ. Однако число этих модулей может варьироваться в системе от 1 до 32. Таким образом, при максимальной комплектации в системе может подвергаться ассоциативной обработке 256 кбайт информации. Скорость поиска и обработки информации 256 процессорными элементами высока, и остальные элементы системы спроектированы так, чтобы поддерживать эту скорость.

Устройство управления ассоциативными модулями организует выполнение операций над данными по командам, хранящимся в управляющей памяти. Оно может выбирать несколько рабочих подмножеств из общего множества данных, хранимых в АМ, и выполнять над этими подсистемами операции, не затрагивая остальную информацию.

Управляющая память разделена на шесть секций: первая (емкостью 612 слов) – память библиотеки подпрограмм; вторая и третья (512 слов) память команд; четвертая (512 слов) – быстродействующий буфер данных; пятая (16384 слов) – основная память; шестая (10720 слов) – область памяти для прямого доступа. Длина одного слова – 32 разряда. Первые четыре секции выполнены на интегральных схемах и имеют высокое быстродействие с длительностью цикла памяти около 200 нс. Вторая и третья секции (память команд) работают попеременно: одна выдает команды в УУ, а другая в это время загружается от страничного устройства и наоборот. Пятая и шестая секции выполнены на ферритовых сердечниках, длительность цикла примерно 1 мкс. При необходимости емкость пятой секции может быть удвоена. Страничное устройство загружает первые три секции памяти информацией из быстродействующего буфера, основной памяти или памяти прямого доступа.

Последовательный контроллер ассоциативной системы является обычной однопроцессорной ЭВМ типа РДР-11 и обеспечивает работу в режиме трансляции и отладки программ; первоначальную загрузку управляющей памяти, связь между оператором и системой; управление программами обработки прерываний по ошибкам, а также программами технической диагностики обслуживания. Последовательный контроллер снабжен памятью (емкость 8 кслов), печатающим устройством, перфоленточным вводом – выводом и имеет интерфейс, обеспечивающий связь с другими элементами системы.

Подсистема ввода – вывода обеспечивает возможность подключения к системе STARAN других вычислительных устройств и разнообразного периферийного оборудования. Имеются четыре вида интерфейсов: прямой доступ к памяти; буферизованный ввод - вывод; параллельный ввод – вывод; логическое устройство внешних функций. Прямой доступ к памяти позволяет использовать память внешней (несистемной) ЭВМ как часть управляющей памяти системы. Эта память становится таким образом доступной как для внешней ЭВМ, так и для системы SPARAN. При этом нет необходимости в буферизации передаваемой между ними информации.

Интерфейс прямого доступа может использоваться и для подключения внешней памяти. Буферизованный ввод – вывод используется для связи системы со стандартными периферийными устройствами, обмен производится блоками данных или программ. Этот интерфейс может использоваться и для связи с несистемной ЭВМ, однако прямой доступ там все-таки предпочтителен,

так как обмен производится быстрее и нет необходимости формирования информации в блоки перед передачей. Параллельный ввод – вывод, который включает в себя по 256 входов и 256 выходов для каждой матрицы, является важной составной частью подсистемы ввода – вывода. Он позволяет увеличить скорость передачи данных между матрицами, обеспечить связь системы с высокоскоростными средствами ввода – вывода и непосредственную связь любого устройства с ассоциативными модулями. С помощью параллельного ввода – вывода можно, в частности, подключать, к ассоциативным матрицам накопители на магнитных дисках, что позволяет быстро вводить и выводить большие объемы информации.

Совокупность всех перечисленных средств, входящих в систему STARAN, позволяет выполнять одновременно сотни и тысячи одинаковых операций при решении определенных классов задач.

3.4. ОДНОРОДНЫЕ СИСТЕМЫ И СРЕДЫ

В предыдущих параграфах был рассмотрен ряд вычислительных систем, при построении которых используются различные возможности по организации параллельной обработки. Во всех этих системах не ставится никаких условий или ограничений относительно состава и функций устройств, а также связей между ними.

В начале 00-х годов Э. В. Евреинов и Ю. Г. Косарев предложили несколько другой подход к построению систем, в основе которого три принципа: параллельность операций; переменность логической структуры; конструктивная однородность элементов и связей между ними.

Первый принцип базируется на аксиоме параллельности задач и алгоритмов: всякая сложная задача может быть представлена в виде связанных между собой простых подзадач и для любой сложной задачи может быть предложен параллельный алгоритм, допускающий ее эффективное решение. Аксиома параллельности, таким образом, обеспечивает достижение высокой производительности за счет параллельной работы большого числа обрабатывающих устройств или элементов.

Второй принцип базируется на аксиоме переменности логической структуры: процесс решения сложной задачи может быть представлен некоторой структурной моделью, включающей в себя подзадачи и связи между ними. Это означает, что для каждой сложной задачи можно предложить соответствующую структуру из обрабатывающих элементов, связанных между собой определенным образом.

Третий принцип базируется на аксиоме конструктивной однородности элементов и связей: все простые задачи получаются путем деления сложной задачи на части, а поэтому все эти простые задачи примерно одинаковы по объему вычисления и связаны между собой одинаковыми схемами обмена. Это означает, что система для решения сложной задачи может быть построена из одинаковых обрабатывающих элементов, связанных между собой одинаковым образом.

Таким образом, при соблюдении трех принципов вычислительная система может быть представлена как совокупность неограниченного числа одинаковых обрабатывающих устройств, однотипно связанных между собой. Эта совокупность перед решением задачи настраивается соответствующим образом. Такая однородная вычислительная система должна решать задачи неограниченной сложности и объема при высокой надежности и готовности, что обеспечивается избыточностью обрабатывающих устройств, унифицированностью связей между ними и легкостью перестройки системы.

В рассмотренных выше комплексах и системах в той или иной степени используются указанные выше три принципа: почти везде возможна параллельная обработка задач; во многих случаях, и первую очередь в МПВК, возможна реконфигурация и иногда прямая настройка системы на решение определенной задачи; наблюдается стремление унифицировать обрабатывающие средства, т. е. создавать по возможности однородные системы.

Описываемый подход к созданию вычислительных систем в виде однородных перестраиваемых структур отличается тем, что используются все три принципа, что, по мнению авторов, и должно обеспечить максимальный эффект.

Хотя эти идеи были выдвинуты более двадцати лет назад, достаточно активная их реализация началась сравнительно недавно. Это и понятно, так как создание однородных систем из большого числа ЭВМ или процессоров второго и даже третьего поколения было вовсе не простой задачей: слишком громоздкими получались такие системы. Ясно, что эффективная реализация возможна лишь при весьма эффективной элементной базе. С развитием микропроцессоров и микро-ЭВМ на их основе такая база создана, и стала возможной реализация этих идей. В качестве иллюстрации можно рассмотреть систему МИНИМАКС [5].

Эта система относится к классу однородных и имеет программируемую макроструктуру. Число элементарных машин (ЭМ) в системе не фиксировано и может определяться классом решаемых задач. Структурную единицу системы составляет ЭМ, которая включает в себя вычислительный модуль (ВМ) и модуль межмашинной связи (ММС). Состав каждой ЭМ также достаточно произвольный – может выбираться из числа типовых спецификаций семейств АСВТ или СМ ЭВМ. В качестве ВМ используются ЭВМ, построенные на основе процессоров М-6000, М-7000, СМ ЭВМ. Модули МС, предназначенные для организации взаимодействия ЭМ, выполнены как автономные устройства, обеспечивают передачу данных, адресов и управляющей информации между соседними ВМ.

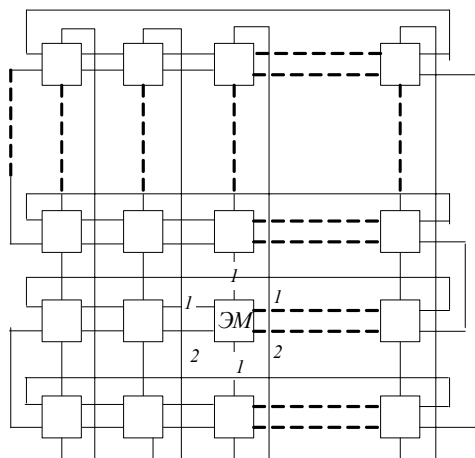


Рис. 3.10. Связи между ЭМ в системе МИНИМАКС

Из рис. 3.10 видно, что каждая элементарная машина связана с четырьмя соседними и взаимодействуют они с помощью двух типов связи: 1 и 2, обеспечивающих полудуплексную работу. Связи типа 1 используются для пересылки данных между ОЗУ передающего ВМ и одного или нескольких принимающих ВМ, передачи адресов между передающими и принимающими ВМ и обмена логическими переменными между элементарными машинами. Связи типа 2 являются вспомогательными: они используются для программирования соединений между элементарными машинами по линиям связи типа 1, а также для передачи управляющей информации, обеспечивающей использование общих ресурсов – периферийных устройств, файлов, программ и др. Связь типа 3 между вычислительным модулем и модулем машинной связи в пределах одной элементарной машины – дуплексные и используются для обмена информацией с периферийными устройствами.

Важным аспектом работы однородной системы является программирование ее структуры – настройка, которая осуществляется с помощью специальных регистров настройки, входящих в состав модулей машинной связи. Содержимое этих регистров может быть изменено либо собственным вычислительным модулем, либо вычислительным модулем любой другой элементарной машины.

Система МИНИМАКС – общего назначения и может работать в автоматизированных системах управления технологическими процессами.

Развитием однородных вычислительных систем являются однородные вычислительные среды, которые представляют собой и в общем случае n -мерную решетчатую структуру. В частом случае это двумерная структура из квадратных клеток, заполняющих плоскость. Каждая клетка в

двухмерной структуре соединяется с четырьмя соседними. Между двумя соседними клетками проходят два канала настройки и один канал передачи рабочей информации (рис. 3.11). Каждый элемент однородной вычислительной среды состоит из коммутационных и функциональных компонентов. Функциональный компонент реализует полную систему логических функций, например: ИЛИ – НЕ – $f(x_1, x_2) = x_1 \bar{V}x_2$ или ИЛИ – НЕ – $f(x_1, x_2) = \bar{x}_1 \bar{x}_2$, Коммутационные и функциональные компоненты позволяют путем соответствующей настройки при достаточном их числе реализовать любую структуру.

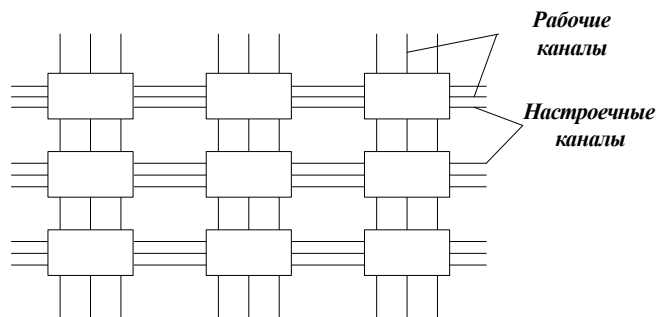


Рис. 3.11. Однородная вычислительная среда

Пока еще нет достаточного опыта, который позволил бы оценить перспективность этих идей, однако можно отметить, что реализация однородных систем – непростая проблема, а эффективность таких систем не является безусловной. Надо иметь в виду, что связи между отдельными элементами достаточно сложны, аппаратные затраты на реализацию связей велики, причем чем элементарнее функции вычислительного модуля, тем больший удельный вес приобретают затраты на связи. Кроме того, программирование связей для настройки системы является достаточно сложной задачей. Вместе с тем не вызывает сомнения, что однородные вычислительные системы представляют значительный интерес и, по-видимому, будут активно развиваться.

3.5. ФУНКЦИОНАЛЬНО РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ

На вычислительные системы общего назначения, используемые для научно-технических расчетов и моделирования, в системах автоматизированного проектирования и управления производством, возлагается выполнение широкого спектра задач. При этом вычислительная система должна реализовать обширный набор функций над различными типами и структурами данных: обработку целочисленных значений, действительных чисел, графической информации и изображений, текстов, матричную обработку, трансляцию программ, доступ к данным, организованным в наборы или базы, и т. д. Кроме того, для управления вычислительными процессами и функционированием необходимо реализовать специфические функции управляющих программ операционной системы, управления виртуальной памятью, сходствами ввода – вывода и передачи данных, а также контроль и диагностику системы и др.

В ЭВМ первого и второго поколений все эти функции реализовались одним процессором и интерпретировались им в виде арифметических и логических операций. Такое же положение сохранилось в основном и в ЭВМ третьего поколения, в которых лишь простейшие операции ввода – вывода снимались с центрального процессора и передавались специализированным устройствам – каналам или процессорам ввода – вывода.

Высокопроизводительные системы общего назначения создаются на основе многопроцессорных комплексов. Использование в таких системах однотипных процессоров, аналогичных процессорам ЭВМ общего назначения, оказывается неэкономичным, поскольку в каждом процессоре в каждый момент времени используется лишь часть ресурсов, обеспечивающих обработку данных одного типа. Наиболее экономичный способ построения многопроцессорной системы общего назначения – использование специализированных процессоров, ориентированных на реализацию определенных функций: обработки

скалярных величин, текстов, матричной обработки, трансляции программ, управления данными и др. При этом значительно сокращаются затраты оборудования в процессоре и повышается его производительность. Кроме того, совокупность таких процессоров предоставляет необходимый для решения задач набор функций, который можно изменять, по-разному комплектуя систему и приспособив ее к рабочей нагрузке.

Многопроцессорные вычислительные системы, построенные на основе разнотипных процессоров, ориентированных на реализацию определенных функций, называются функционально распределенными вычислительными системами (ФРВС). Это неоднородные системы и строятся они как проблемно-ориентировочные – путем включения в их состав набора процессоров, соответствующего потребностям обрабатываемых задач.

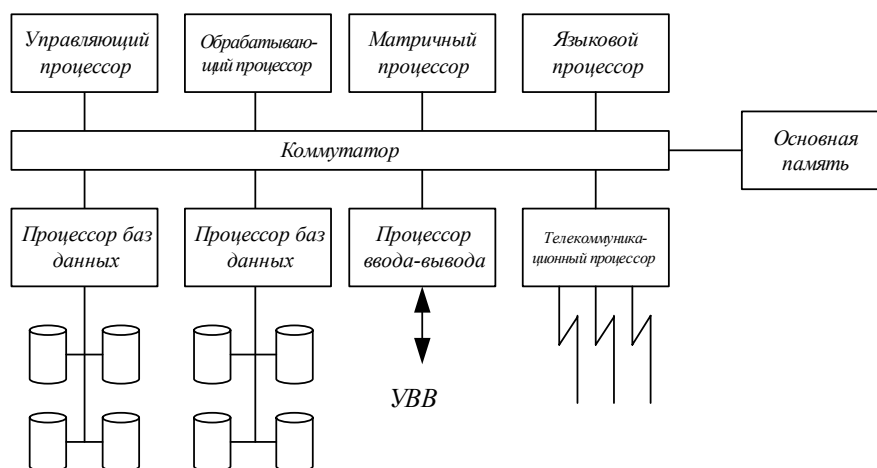


Рис. 3 12 Функционально распределенная вычислительная система

Структура и функционирование. Принцип структурной организации ФРВС представлен на рис. 3.12. Система состоит из совокупности процессоров, имеющих индивидуальную память и основной памяти. Ядро системы обеспечивает информационное сопряжение всех устройств. Ядро может быть реализовано в виде, системной шины (магистральной), коммутационного поля или коммутатора основной памяти. В первых двух случаях каждый процессор может обмениваться данными с любыми другими процессорами и основной памятью. При использовании коммутатора основной памяти обмен данными производится только через память. В представленной структуре управляющий процессор реализует супервизорные функции – управление ресурсами и задачами, обрабатывающий процессор – обработку числовых и символьных данных, матричный процессор – матричную и векторную обработку, языковой процессор – трансляцию программ, процессоры баз данных – доступ к наборам данных и управление базами данных, процессор ввода – вывода обслуживает устройства ввода – вывода и телекоммуникационный процессор обеспечивает передачу данных по каналам связи. Состав процессоров в конкретной системе зависит от класса решаемых задач. Так, в системе могут использоваться два обрабатывающих процессора или несколько телекоммуникационных.

Обработка каждой задачи распределяется между процессорами. При этом функции управления данными реализуются процессорами, изображенными в нижней части рисунка. Разные шаги заданий, программы и ветви (блоки) программ выполняются обрабатывающим, матричным и языковым процессорами. Распределение ресурсов между задачами и управление задачами производится управляющим процессором, который реализует управляющие программы операционной системы. Загрузка оборудования увеличивается за счет мультипрограммирования и, возможно, параллельных вычислений на уровне подзадач.

Специализация процессоров обеспечивается на разных уровнях – на уровне структуры, микропрограммном и программном. Специализация на уровне структуры достигается за счет использования и операционной части процессора специальных

регистровых структур и микрооперации, эффективно реализующих заданный набор операций. Такими являются матричные процессоры, содержащие совокупность арифметическо-логических устройств, с помощью которых параллельно обрабатываются векторы и матрицы. Специализация на микропрограммном уровне сводится к созданию с помощью микропрограмм специализированного набора операций, ориентированного на вычисление заданного набора функций. При использовании ОЗУ для хранения микропрограмм специализация процессора производится путем динамического микропрограммирования – загрузки в память соответствующего набора микропрограмм. В этом случае возможно оперативно изменять конфигурацию системы, загружая в структурно одинаковые процессоры необходимые наборы микропрограмм. Функциональная специализация процессоров на программном уровне достигается за счет загрузки в процессор соответствующего набора программ.

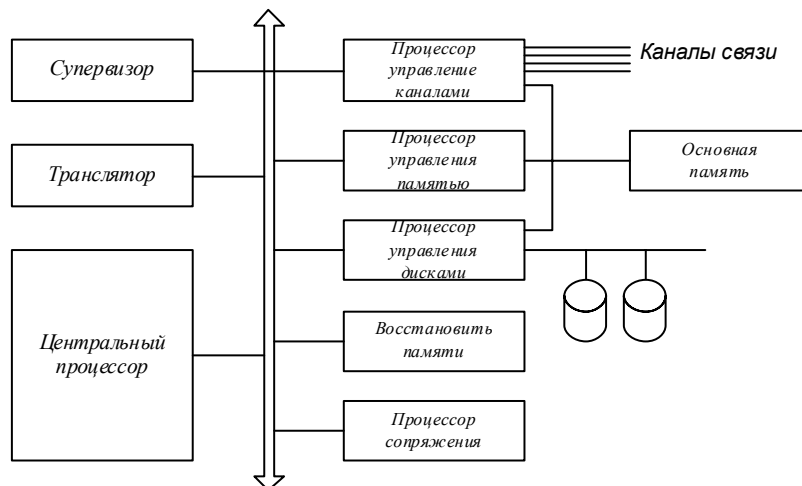


Рис. 3.13. Система SYMBOL

В ФРВС используются, как правило, все три уровня специализации процессоров. Обрабатывающие и матричные процессоры имеют специализированную структуру: первые – для выполнения традиционных операций над логическими значениями, целыми и действительными числами и строками символов, последние – для производства векторных и матричных операций. Остальные процессоры функционально специализируются на уровне микропрограмм или программ.

Реализация. Принцип функционально распределенной организации применялся уже в ЭВМ третьего поколения, где для ввода – вывода использованы каналы и процессоры ввода – вывода. Примерами являются ЭНМ серий IBM 360, IBM 370 и ЕС ЭВМ.

Одна из первых систем, в полной мере реализующая принцип функционально распределенной организации, – система SYMBOL, созданная в 1970 г. Это неоднородная восьмипроцессорная система (рис. 3.13). Процессор-супервизор управляет работой всей системы, координируя остальные процессоры, создавая очереди заявок к ним и распределяя процессоры между задачами. Процессор-транслятор обеспечивает перевод операторов с входного языка на внутренний язык системы (на машинный язык). Центральный процессор реализует обычные функции выборки команд и также арифметические и логические операции. В системе используется виртуальная память, работа которой обеспечивается процессором управления памятью. Этот процессор обрабатывает заявки от других процессоров на запись и чтение данных произвольной структуры. Виртуальная память состоит из 2^{16} страниц, содержащих 256 64-битных слов, емкость основной памяти – 8 кслов (32 страницы). В дисковой памяти размещается примерно 50 тыс. страниц. Распределение емкости дисковой памяти, поиск и передача требуемой страницы данных реализуются процессором управления дисками. Управление вводом – выводом данных возложено на процессор управления каналами. К этому процессору через каналы передачи данных подключены внешние устройства. Редактирование и

преобразование вводимых – выводимых данных обеспечивается процессором сопряжения, который работает в основном совместно с процессором управления каналами.

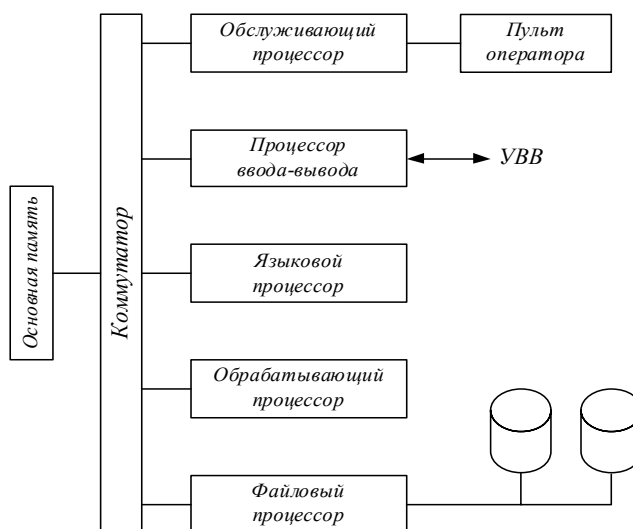


Рис. 3.14. Система System/80

Процессоры сопрягаются посредством главной шины, состоящей из 111 линий причем используются для передачи слова данных 64 линии, адреса слова – 24 линии, адреса абонента, которому направляются данные, – 5 линий. Остальные линии служат для передачи кода операции, приоритета сообщений и для синхронизации работы абонентов. При относительно низком быстродействии (длительность цикла процессоров – 320 нс и оперативной памяти – 2,5 мкс) система отличается высокой производительностью, составляющей 75 тыс. операторов входного языка в минуту, что примерно в 10 раз больше, чем у больших ЭВМ общего назначения.

На рис 3 14 представлена упрощенная структурная схема вычислительной системы семейства System/80 фирмы IBM. В системе может использоваться до 8–16 процессоров, взаимодействующих через основную память и интерфейс прямого управления. Обслуживающий процессор обеспечивает работу пульта системы, планирование, контроль и диагностику. Процессор ввода – вывода обслуживает внешние устройства и выполняет первичную обработку (редактирование) вводимых – выводимых данных. Языковой процессор предназначен для трансляции программ с языка высокого уровня на машинный язык, т. е. в систему команд соответствующих процессоров. Обрабатывающий процессор выполняет обычные операции центрального процессора ЭВМ. Файловый процессор управляет данными, реализуя создание, открытие и закрытие наборов данных и доступ к данным, хранимым в наборах с различной организацией. Каждый процессор имеет собственную оперативную память. Функциональная ориентация процессоров обеспечивается на микропроцессорном уровне – путем загрузки в процессор соответствующих наборов микропрограмм. Возможности системы могут расширяться за счет подключения нескольких процессоров ввода – вывода, обрабатывающих процессоров, а также матричного процессора.

3.6. СИСТЕМЫ С ПЕРЕСТРАИВАЕМОЙ СТРУКТУРОЙ

Универсальный способ создания высокопроизводительных и высоконадежных вычислительных систем – объединение ЭВМ (процессоров) в многомашинные (многопроцессорные) комплексы, обеспечивающие

- 1) параллелизм процессоров управления, доступа к данным и обработки;
- 2) распределенность процессоров управления, доступа к данным и обработки между модулями системы, т. е. децентрализованность управления работой системы и асинхронность взаимодействия процессоров и модулей;
- 3) перестраиваемость структуры с целью адаптации системы к потребностям задач в ресурсах и повышения устойчивости к отказам элементов;

- 4) открытость, т. е. возможность развития системы за счет подключения к ней дополнительных модулей без изменения принципов функционирования имеющихся модулей;
- 5) модульность технических и программных средств и регулярность (в пределе – однородность) структуры.

Параллелизм вычислительных процессов и процессов управления создает основу для повышения производительности системы. Распределенность процессов позволяет строить высокопроизводительные системы из достаточно простых модулей, например из микро-ЭВМ с относительно небольшим быстродействием и ограниченной емкостью памяти. Перестраиваемость структуры обеспечивает, с одной стороны, высокую производительность системы за счет ее адаптации к вычислительным процессам и составу обрабатываемых задач и, с другой стороны, живучесть системы при отказах элементов. Открытость системы позволяет в рамках фиксированной архитектуры создавать системы разной производительности за счет изменения числа модулей от единиц до десятков, сотен и, возможно, тысяч. Модульность технических и программных средств существенно упрощает разработку и производство элементов системы, за счет чего снижается ее стоимость, а также порождает регулярность структуры и, следовательно, упрощает управление системой (процессами и ресурсами) и ее эксплуатацию.

В последние десятилетия ведутся интенсивные исследования в области создания параллельных распределенных открытых многомодульных систем с перестраиваемой структурой, на основе которых разработано большое число экспериментальных и рабочих систем повышенной производительности и надежности. Архитектура систем с рассматриваемыми свойствами имеет особое значение для использования микро-ЭВМ в качестве элементной базы. Возможность неограниченного объединения микро-ЭВМ в системе, эффективно адаптирующиеся к потребностям задач, позволила бы решить многие проблемы, в том числе обеспечить пользователей высокопроизводительными средствами обработки числовых данных, графической информации и изображений. Однако для создания таких систем необходимо решить комплекс проблем системного управления, представляющих собой программно-аппаратурную надстройку над базовыми микро-ЭВМ. Среди этих проблем важнейшими являются:

- 1) структурная организация систем, обеспечивающая образование ансамблей процессоров, запоминающих устройств и каналов обмена данными, соответствующих потребностям вычислительного процесса, при умеренных затратах ресурсов на их организацию и координацию;
- 2) организация вычислительных процессов, обеспечивающая их параллелизм, распределение по системе модулей и координацию асинхронно выполняемых подпроцессов при умеренных издержках;
- 3) создание языков высокого уровня, описывающих алгоритмы в системно-независимой форме и сохраняющих представление о параллелизме вычислительного процесса.

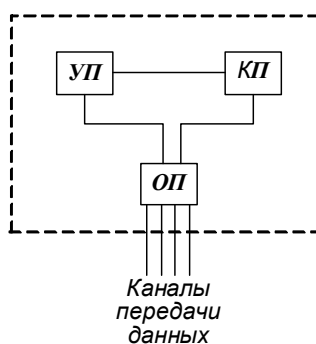


Рис. 3.15. Состав модуля системы с перестраиваемой структурой

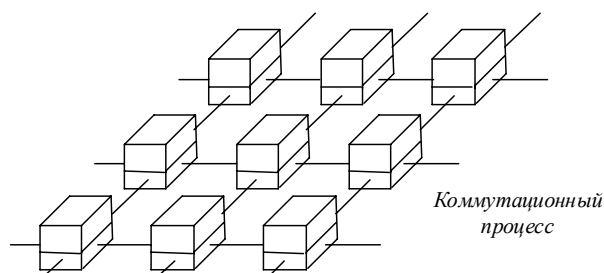


Рис. 3.16. Фрагмент матричной структуры

К настоящему времени ли проблемы не решены окончательно и известные разработки вычислительных систем с перестраиваемой структурой лишь отчасти обладают требуемыми свойствами.

Структурная организация. Вычислительные системы с перестраиваемой структурой строятся на основе микропроцессорных модулей. Модуль должен реализовать следующие функции: 1) обработку данных, сводящуюся к обработке логических значений, числовых значений, представленных в виде целых и действительных чисел, и строк символов; 2) управление вычислительным процессом, обеспечивающее взаимодействие модуля с ансамблем модулей, реализующих процесс, и с системой в целом; 3) установление соединений с другими модулями и передачу данных между ними для обеспечения вычислительных процессов. С учетом указанных функций модуль вычислительной системы рассматривается как совокупность трех процессоров (рис. 3.15): обрабатывающего (ОП), управляющего (УП) и коммутационного (коммуникационного) (КМ). Коммутационный процессор обеспечивает обслуживание нескольких (обычно двух – шести) каналов передачи данных. Физически модуль может реализоваться на основе одной микро-ЭВМ, выполняющей в мультипрограммном режиме функции обработки, управления процессами и передачи данных, или на основе нескольких микропроцессоров, между которыми разделяются выше перечисленные функции.

В вычислительных системах с перестраиваемой структурой модули объединяются в простейшие структуры, позволяющие достаточно легко определять пути соединения между взаимодействующими модулями. Наиболее подходящими для построения рассматриваемых систем являются матричные, пирамидальные и кубические структуры. Фрагмент матричной структуры изображен на рис. 3.10. Модули, в которых выделен коммутационный процессор, соединяются посредством последнего в матрицу. Коммутационные процессоры и каналы связи образуют в совокупности коммутационное поле, обеспечивающее соединение взаимодействующих модулей и передачу данных между ними. Часть модулей системы специализируется на обслуживании периферийных устройств – накопителей на магнитных дисках и лентах и устройств ввода – вывода. Управляющие и коммутационные процессоры модулей ввода – вывода обеспечивают взаимодействие внешних устройств с процессами, реализуемыми в любом из модулей системы.

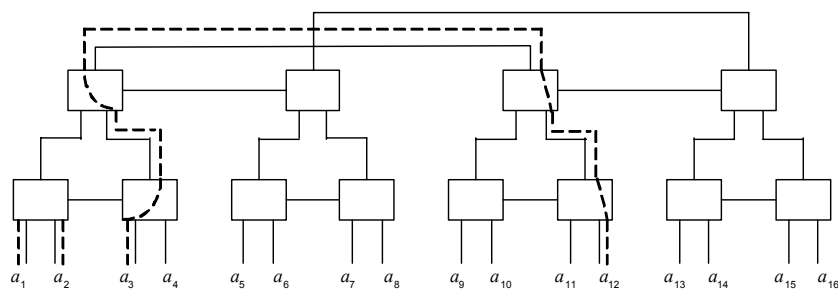


Рис. 3.17. Многоуровневое коммутационное поле

Другой способ структурной организации системы с перестраиваемой структурой на основе вычислительных комплексов, имеющих коммутационное поле, которое образовано совокупностью коммутаторов с децентрализованным управлением. Наиболее экономичными являются поля с многоуровневой организацией, пример которых приведен на рис. 3.17. Здесь a_1, \dots, a_{16} входы коммутатора, к которым подсоединяются микропроцессорные модули. Штриховыми линиями показаны примеры соединения модулей через коммутатор. В таких полях нижний уровень коммутации обеспечивает соединения между соседними модулями, образующими отдельные группы, следующий уровень коммутации – соединения между соседними группами и т. д. Многоуровневые коммутационные поля позволяют создавать соединения между любыми парами процессорных модулей с помощью умеренного числа коммутационных модулей в системах с матричной, кубической и пирамидальной структурами. Однако среднее число одновременно устанавливаемых соединений в многоуровневых коммутационных полях меньше среднего числа соединений, обеспечиваемых матричными и аналогичными структурами, что приводит к снижению степени параллелизма вычислительных процессов и, следовательно, к уменьшению производительности системы.

Организация вычислительных процессов. Основные проблемы организации вычислений в системах с перестраиваемой структурой связаны с обеспечением параллелизма вычислений и распределенного децентрализованного управления процессами и ресурсами. Эти проблемы разработаны только в первом приближении, и известные способы организации параллельных вычислений в распределенных системах с децентрализованным управлением еще не достигли необходимого уровня универсальности и формализации.

Параллельная обработка задач, т. е. мультипрограммный режим функционирования системы, обеспечиваемая достаточно простыми средствами. После ввода задания в систему модуль, принявший задание, посылает через коммутационное поле запрос на поиск свободного обрабатывающего модуля. Когда свободный модуль найден, ему посылается задание, определяющее имена наборов данных, в которых размещается программа, исходные данные и в которые должны быть помещены результаты вычислений. Из задания и программы модуль получает сведения о ресурсах, необходимых для выполнения задания: емкости операционной памяти, числе процессоров и неразделяемых наборах данных. Модуль закрепляет за собой необходимые ресурсы, и после обеспечения задания требуемыми ресурсами инициируется процесс выполнения задачи. По завершении обработки ресурсы освобождаются и в дальнейшем предоставляются очередным заданиям. Число процессов, реализуемых параллельно, определяется числом модулей, входящих в состав системы, и при наличии очереди заданий производительность системы пропорциональна числу модулей.

Параллельные программы строятся традиционными способами: выделением подзадач и ветвей программы, операций над векторами и матрицами и организацией конвейерной обработки данных. Наиболее просто реализуются вычисления с выделением подзадач и параллельных ветвей. При возникновении ветви в ведущей программе модуль посылает запрос на поиск свободного модуля, в который загружается программа и данные ветви, и ветвь выполняется как самостоятельная задача, по завершении которой в ведущий модуль отсылаются результаты обработки. Параллельные вычисления по конвейерной и матричной схемам организуются за счет создания соответствующих конфигураций связей между модулями – линейных (кольцевых) и матричных структур. Построение таких структур в многомодульных системах, в которых часть модулей занята выполнением ранее созданных задач, является пока нерешенной проблемой. Обычно для матричных вычислений в систему встраивает в качестве специального модуля матричный процессор, обеспечивающий высокопроизводительную обработку блоков данных.

В вычислительной системе с перестраиваемой структурой должно быть реализовано распределенное (децентрализованное) управление ресурсами. Это означает, что в системе не должно быть выделенного модуля (даже многократно зарезервированного), на который возложена задача централизованного управления функционированием системы.

Распределенное управление основано на согласованной работе всех модулей системы, каждый из которых реализует одинаковый набор правил управления, обеспечивающий эффективное использование всех ресурсов системы. Распределенное управление повышает надежность системы, поскольку каждый модуль способен реализовать управление ресурсами и процессами, и одновременно повышает производительность системы, так как управляющие решения формируются без затрат времени на сбор информации о состоянии всех элементов системы (а за это время ситуация в системе может существенно измениться).

Как в любой многопроцессорной системе, механизм управления должен исключать взаимную блокировку процессов при запросах ресурсов. Такая ситуация возникает, если процесс *A* располагает ресурсом *a* и требует для своего исполнения ресурс *b*, а процесс *B* располагает ресурсом *b* и дополнительно нуждается в ресурсе *a*. Например, процесс *A* располагает модулем *a* и требует соединения *b* с ним, а процесс *B* располагает соединением *b* и требует модуль *a*. Для предотвращения блокировок используются различные механизмы управления ресурсами, одновременное формирование запросов на все необходимые ресурсы и освобождение всех ресурсов, если не выделен хотя бы один из них; разделение ресурсов по типам и иерархический порядок выделения ресурсов и т. д. Кроме того, механизм управления должен принимать решения о передислокации программ и наборов данных между модулями, о необходимости подключения к процессу дополнительных модулей или последовательной реализации алгоритмически параллельных процессов на одном модуле и, наконец, о распределении задач между модулями, обеспечивающем необходимое время решения и высокую производительность системы. Для управления процессами обычно предлагаются эвристические процедуры, не требующие большой емкости памяти и трудоемких вычислений. Однако эффективность большинства предлагаемых процедур к настоящему времени не оценена в достаточной степени.

СИСТЕМЫ ТЕЛЕОБРАБОТКИ

4.1. ПРИНЦИПЫ ПОСТРОЕНИЯ

Система телеобработки – совокупность технических и программных средств, предназначенная для обработки на ЭВМ данных, передаваемых по каналам связи. Типичная конфигурация системы телеобработки изображена на рис. 4.1. Абоненты системы (пользователи, технические объекты) подключаются к ЭВМ с помощью каналов связи. Канал связи состоит из линии связи, по которой передаются сигналы, и аппаратуры передачи данных (АПД), преобразующей данные в сигналы, соответствующие типу линии связи (канала). Состав канала связи представлен на рис. 4.2, где ООД – окончательное оборудование данных, передающее и принимающее последовательности битов, составляющих данные.

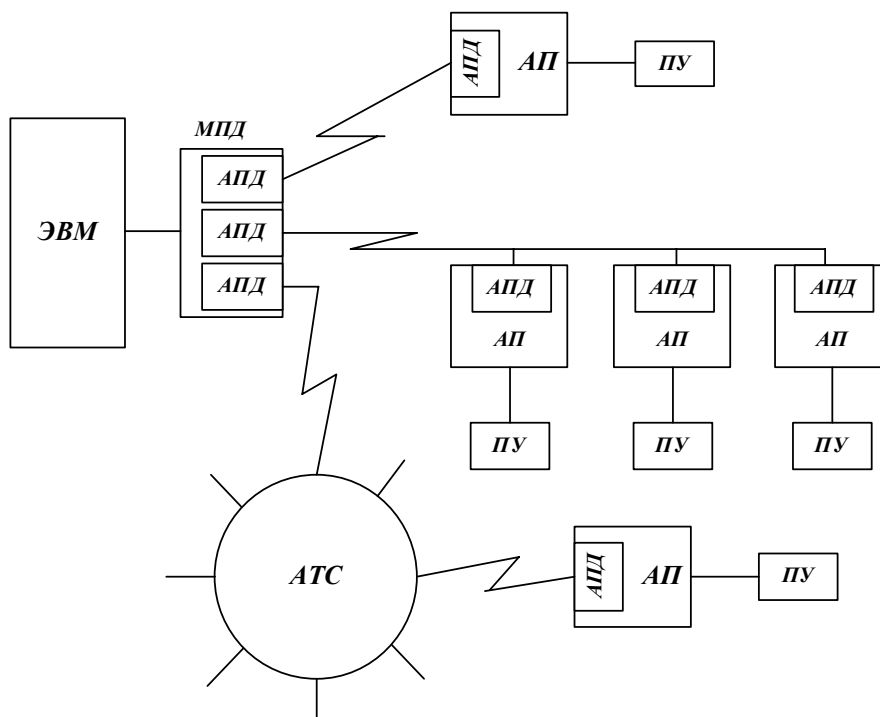


Рис. 4.1. Средства телеобработки данных



Рис. 4.2. Состав канала данных

Абоненты взаимодействуют с ЭВМ через абонентские пункты (АП) – терминальные устройства системы телеобработки. Абонентский пункт содержит в своем составе АПД, обслуживающую канал связи, набор периферийных устройств (ПУ), используемых для ввода – вывода данных, и обеспечивает обмен данными между каналом связи и периферийными

устройствами. В общем случае для подключения абонентов к ЭВМ используется значительное число каналов связи, которые подключаются к ЭВМ через мультиплексор передачи данных (МПД), содержащий средства для обмена данными между ЭВМ и АПД каналов связи, т. е. для ввода – вывода данных по каналам связи. Функционирование технических средств системы телеобработки – мультиплексоров, каналов связи и абонентских пунктов – поддерживается программными средствами телеобработки, реализуемыми ЭВМ, абонентскими пунктами и, возможно, мультиплексорами передачи данных.

В системах телеобработки используются различные конфигурации связей между ЭВМ и абонентами, зависящие от состава и схемы размещения абонентов, типа используемых каналов связи и интенсивности потока данных между абонентами и ЭВМ. Наиболее широко используются выделенные каналы связи (некоммутируемые каналы), закрепленные за ЭВМ и соответствующими абонентами. Канал может обслуживать единственного абонента, образуя двухточечное соединение (верхний канал на рис. 4.1), или одновременно нескольких абонентов (средний канал на рис. 4.1), образуя многоточечное (многopунктовое) соединение. В последнем случае абонентские пункты разделяют между собой канал во времени, принимая адресованные им данные и снабжая передаваемые данные адресом источника. Для подключения абонентов могут использоваться каналы сетей связи общего применения сетей автоматической телефонной и телеграфной связи (нижний канал на рис. 4.1). В этом случае соединение между абонентом и ЭВМ является коммутируемым и устанавливается, например, набором номера вызываемого абонента.

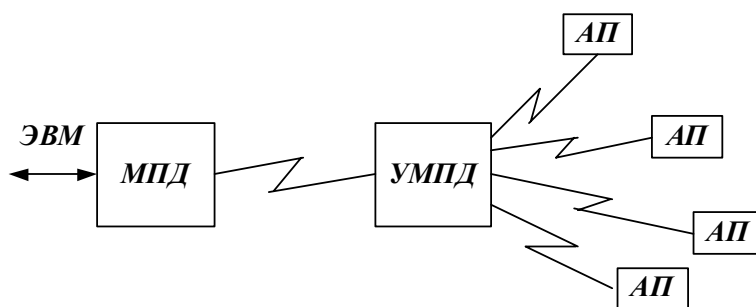


Рис. 4.3. Подключение абонентов через удаленный МПД

Передача данных по высокоскоростным ($5 \cdot 10^3 - 10^5$ бит/с) каналам связи обходится значительно дешевле, чем по совокупности низко- и среднескоростных каналов связи, имеющих пропускную способность $10^2 - 5 \cdot 10^5$ бит/с. Поэтому группы удаленных абонентов подключаются к ЭВМ по схеме, изображенной на рис. 4.3. Для связи используется высокоскоростной канал, к которому на удаленном конце подключен удаленный мультиплексор передачи данных (УМПД). Последний соединяется низкоскоростными каналами с АП и обеспечивает коллективное использование высокоскоростного канала многими абонентскими пунктами, работающими в режиме временного разделения (мультиплексирования) высокоскоростного канала.

Основная цель создания систем телеобработки – обеспечить прием данных непосредственно с мест их появления и выдачу результатов обработки к местам их использования. За счет этого существенно повышается эффективность обработки данных: отпадает необходимость в промежуточных носителях данных (перфоленты, перфокарты и магнитные ленты) и повышается оперативность взаимодействия с ЭВМ. В результате этого повышается эффективность работы системы, для которой производится обработка данных. Например, только на основе телеобработки могут строиться эффективные системы управления воздушным и другими видами транспорта и системы автоматизированной продажи билетов.

Кроме того, телеобработка позволяет эффективно использовать мощные ЭВМ, на основе которых можно создавать большие базы данных, что, в свою очередь, снижает стоимость обработки данных, поскольку стоимость миллиона процессорных операций уменьшается с увеличением быстродействия ЭВМ. С помощью линий связи к таким ЭВМ подключается значительное число терминалов и, следовательно, пользователей, что обеспечивает высокий уровень загрузки ЭВМ и ее постоянство во времени. Таким образом, телеобработка расширяет сферу применения ЭВМ и позволяет повысить эффективность системы обработки данных.

Для создания систем телеобработки необходимы следующие технические средства: 1) каналы связи, в том числе и аппаратура передачи данных; 2) устройства сопряжения ЭВМ с АПД; 3) абонентские пункты; 4) удаленные мультиплексоры передачи данных.

Значительная протяженность линий связи исключает возможность обмена отдельными сигналами между ЭВМ и оконечным оборудованием, во-первых, из-за технических трудностей передачи отдельных сигналов и, во-вторых, из-за помех, воздействующих на каналы связи и искажающих сигналы. Поэтому взаимодействие ЭВМ и оконечного оборудования организуется с помощью сообщений – блоков данных, передаваемых в виде единого целого. Сообщения имеют специальную структуру, обеспечивающую представление и них наряду с собственно данными служебной информации, необходимой для идентификации сообщения и защиты данных от искажений. Возможность взаимодействия абонентов с ЭВМ только посредством сообщений вносит определенную специфику в организацию программного обеспечения телеобработки.

4.2. КАНАЛЫ СВЯЗИ

Основные характеристики канала связи (рис. 4.2) – пропускная способность и достоверность передачи данных. *Пропускная способность канала* оценивается предельным числом бит данных, передаваемых по каналу за единицу времени, и измеряется в бит/с (c^{-1}). Достоверность передачи данных характеризуется вероятностью искажения бита, которая для каналов связи без дополнительных средств защиты от ошибок составляет, как правило, 10^{-4} – 10^{-6} . Основная причина искажений – воздействие помех на линию связи и, отчасти, наличие шумов в АПД. Помехи носят импульсный характер и имеют тенденцию к группированию – образованию пачек помех, искажающих сразу группу соседних бит в передаваемых данных.

Линии связи. Для передачи данных используются линии связи различных типов: проводные (воздушные), кабельные, радиорелейные, волоконно-оптические и радиоканалы наземной и спутниковой связи. Кабельные линии состоят из скрученных пар проводов или коаксиальных кабелей. Основные характеристики линий связи – полоса частот, удельная стоимость и помехоустойчивость. Полоса частот $F = f_n - f_g$ определяет диапазон частот f_n, f_g , где f_n и f_g – нижняя и верхняя граница частот, эффективно передаваемых по линии. Полоса частот зависит от типа линии и ее протяженности. Проводные линии связи имеют полосу частот примерно 10 кГц, кабельные – 10^2 кГц, коаксиальные – 10^2 МГц, радиорелейные – 10^3 МГц и волоконно-оптические – 10^2 МГц. Для передачи данных используется коротковолновая радиосвязь с диапазоном частот от 3 до 30 МГц. Удельная стоимость линии определяется затратами на создание линии протяженностью 1 км. Для передачи данных на небольшие расстояния используются в основном низкочастотные проводные линии, на большие расстояния – высокочастотные линии: коаксиальные кабели, волоконно-оптические и радиорелейные линии. Радиосвязь применяется для организации как местной, так и дальней связи. Помехоустойчивость линии зависит от мощности помех, создаваемых в линии внешней средой или возникающих из-за шумов в самой линии. Наименее помехоустойчивыми являются радиолнии, хорошей помехоустойчивостью обладают кабельные линии и отличной – волоконно-оптические линии, не восприимчивые к электромагнитному излучению.

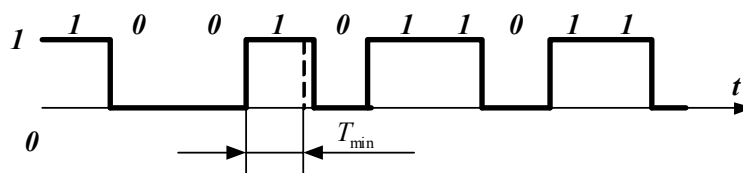


Рис. 4.4. Последовательность двоичных сигналов

Пропускная способность канала. Пропускная способность канала зависит от полосы частот линии связи и отношения мощностей сигнала и шума. Максимальная пропускная способность канала, настроенного на основе линии с полосой частот F и отношением сигнал-шум P_c/P_u , составляет (бит в секунду)

$$C_{\max} = F \log_2 (1 + P_c / P_u) \quad (4.1)$$

Значение $(1 + P_c/P_u)$ определяет число уровней сигнала, которое может быть воспринято приемником. Так, если отношение $P_c/P_u > 3$, то единичный сигнал может переносить четыре значения, т. е. $\log_2(1 + 3) = 2$ бита информации.

При передаче данных широко используются двоичные сигналы, принимающие значения 0 и 1. Временная диаграмма последовательности таких сигналов, передаваемых по линии связи, изображена на рис. 4.4, где сверху указаны значения, переносимые сигналом. Минимальная длительность такта, с которым могут передаваться сигналы по каналу с полосой частот F , равна $T_{\min} = 1/(2F)$. Если вероятность искажения символов 0 и 1 из-за помех одинакова и равна p , то число двоичных символов, которые можно безошибочно передать по каналу в секунду,

$$C = 2F [1 + p \log_2 p + (1 - p) \log_2 (1 - p)] \quad (4.2)$$

Это выражение определяет пропускную способность двоичного канала. Величина в квадратных скобках определяет долю двоичных символов, которые передаются по каналу с частотой $2F$ без искажений. Если помехи отсутствуют, вероятность искажения символа $p = 0$ и пропускная способность $C = 2F$; если вероятность искажения $p = 0,5$, то пропускная способность $C = 0$. Если по каналу передается сообщение длиной n двоичных символом, то вероятность появления в нем точно l ошибок $P(n, l) = C_n^l p^l (1 - p)^{n-l}$, среднее число ошибок $a = np$ и среднее квадратическое отклонение $\sigma = \sqrt{np(1 - p)}$.

Наиболее распространенный тип канала – телефонный с полосой пропускания 3,1 кГц и диапазоном частот от $f_H = 0,3$ кГц до $f_H = 3,4$ кГц. Коммутируемый телефонный канал обеспечивает скорость передачи данных $C = 1200$ бит/с, а некоммутируемый – до 9600 бит/с.

Эффективность использования канала связи для передачи данных принято характеризовать удельной пропускной способностью $B = C/F$, т. е. пропускной способностью на 1 Гц полосы частот канала. Для коммутируемых телефонных каналов удельная пропускная способность не превышает 0,4 бит/(с · Гц), а для некоммутируемых составляет, как правило, 3–5 бит/(с · Гц).

Стандартизованы следующие скорости передачи данных по каналам связи: 200, 300, 600, 1200, 2400, 4800, 9600, 12000, 24000, 48000 и 96000 бит/с. Каналы с пропускной способностью до 300 бит/с называются низкоскоростными, от 600 до 4800 бит/с – среднескоростными и с большей пропускной способностью – высокоскоростными.

Способы передачи данных. Для передачи данных по каналам с различными характеристиками используются разные способы, обещающие максимальное использование свойств

каналов для повышения скорости и достоверности передачи данных при умеренной стоимости аппаратуры.

Данные первоначально предоставляются последовательностью прямоугольных импульсов (рис. 4.4). Для их передачи без искажения требуется полоса частот от нуля до бесконечности. Реальные каналы имеют конечную полосу частот, с которой необходимо согласовать передаваемые сигналы. Согласование обеспечивается, во-первых, путем модуляции – переноса сигнала на заданную полосу частот и, во-вторых, путем кодирования – преобразовании данных в вид, позволяющий обнаруживать и исправлять ошибки, возникающие из-за помех в канале связи.

При использовании высокочастотных проводных и кабельных линий, полоса частот которых начинается примерно от нуля, сигналы можно передавать в их естественном виде – без модуляции (в первичной полосе частот). Каналы, работающие без модуляции, называются телеграфными и обеспечивают передачу данных со скоростью, как правило, 50-200 бит/с.

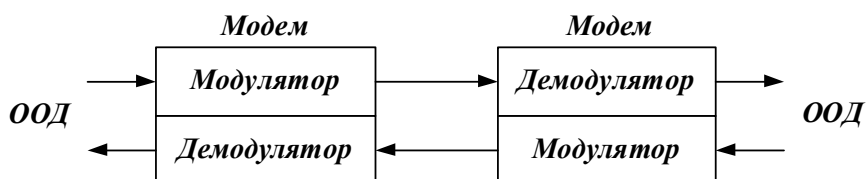


Рис. 4.5. Канал с модуляцией

Когда канал имеет резко ограниченную полосу частот, как, например, радиоканал, передача сигналов должна выполняться в этой полосе и перенос сигнала в заданную полосу производится посредством модуляции по схеме, изображенной на рис. 4.5. В этом случае между окончательным оборудованием данных, работающим с двоичными сигналами, и каналом устанавливается *modem* – модулятор и демодулятор. *Модулятор* перемещает спектр первичного сигнала в окрестность несущей частоты f_0 . *Демодулятор* выполняет над сигналом обратное преобразование, формируя из модулированного сигнала импульсный двоичный сигнал.

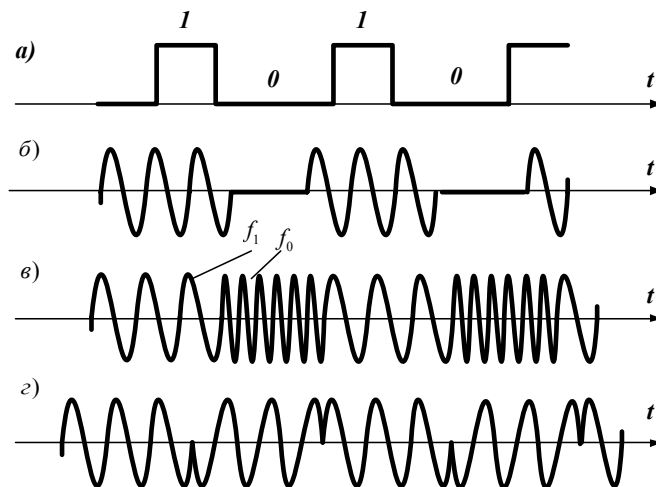


Рис. 4.6. Способы модуляции

Способы модуляции подразделяются на *аналоговые* и *дискретные*. К аналоговым относятся амплитудная, частотная и фазовая модуляция (рис. 4.6). При *амплитудной* (рис. 4.6, б) производится модуляция амплитуды несущей частоты первичным сигналом (рис. 4.6, а). При *частотной модуляции* (рис. 4.6, в) значения 0 и 1 двоичного сигнала передаются сигналами с различной частотой – f_0 и f_1 . При *фазовой модуляции* (рис. 4.6, з) значениям сигнала 0 и 1 соответствуют сигналы частоты f_0 с разной фазой. Дискретные способы модуляции применяются для преобразования аналоговых сигналов, например речевых, в цифровые. Для этих целей наиболее широко используются амплитудно-импульсная, кодово-импульсная и времяимпульсная модуляция.

Кодирование передаваемых данных производится в основном для повышения помехоустойчивости данных. Так, первичные коды символов могут быть представлены в помехозащищенной форме – с использованием кодов Хемминга, обеспечивающих обнаружение и

исправление ошибок в передаваемых данных. В последнее время функция повышения достоверности передаваемых данных возлагается на оконечное оборудование данных и обеспечивается за счет введения информационной избыточности в передаваемые сообщения.

Аппаратура передачи данных. Основное назначение АПД – преобразование сигналов, поступающих с оконечного оборудования, для передачи их в полосе частот канала связи и обратное преобразование сигналов, поступающих из канала. При работе с телеграфным каналом, сигналы по которому передаются без модуляции (в первичной полосе частот), указанные функции реализуются устройством преобразования телеграфных сигналов, а при работе с телефонным и высокочастотным каналом – модемом. Основные элементы модулятора и демодулятора представлены на рис. 4.7. В рассматриваемом случае передача данных в канал производится синхронно с частотой, соответствующей скорости работы канала, например с частотой 1200 Гц. Сигналы синхронизации S_T формируются в модуляторе тактовым генератором $ТГ$. По каждому сигналу синхронизации S_T в блок модуляции $БМ$ вводится двоичный сигнал T , представляющий собой бит данных. Несущая частота формируется генератором $ГНЧ$. Модулированный сигнал поступает на полосовой фильтр $ПФ$, ограничивающий полосу частот сигнала в соответствии с нижней и верхней границей полосы канала. Затем сигнал с заданной полосой частот передается по каналу в демодулятор, проходит через полосовой фильтр, выделяющий заданную полосу частот, и поступает в блок демодуляции.

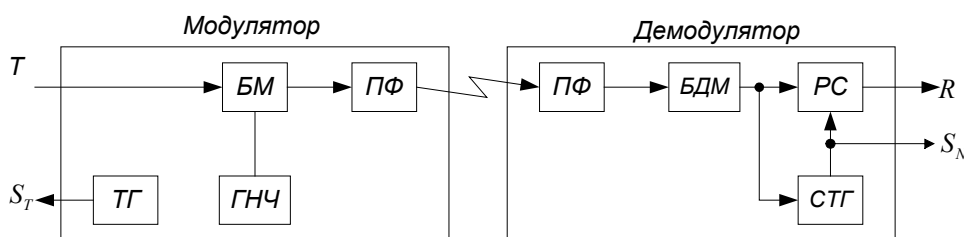


Рис. 4.7. Модулятор и демодулятор

$БДМ$, на входе которого формируются двоичные сигналы. Эти сигналы используются для выделения тактовой частоты, с которой передаются данные. Тактовая частота в демодуляторе формируется синхронизируемым тактовым генератором ($СТГ$), фаза и частота которого автоматически подстраиваются под фазу и частоту сигналов, поступающих с $БДМ$. Сигналы синхронизации S_R поступают на регенератор сигналов $РС$, который формирует прямоугольные импульсы, представляющие собой биты данных со значением 1, и, кроме того, используются аппаратурой обработки данных для синхронизации приема данных из демодулятора. Диаграммы сигналов в демодуляторе представлены на рис. 4.8.

Описанный канал связи называется *синхронным*. В нем передача и прием данных производится с постоянной тактовой частотой, одинаковой на входе и выходе канала. Синхронизм передающего и принимающего оборудования канала обеспечивается автоматически за счет подстройки частоты генератора тактов в демодуляторе. Принимаемые сигналы сдвинуты относительно передаваемых по фазе на величину, определяемую временем распространения сигнала по каналу связи.

В зависимости от направления передачи данных каналы подразделяются на симплексные, полудуплексные и дуплексные. *Симплексный канал* позволяет передавать данные только в одном направлении – прямом или обратном – один абонент передает, а другой принимает данные. *Полудуплексный канал* обеспечивает поочередную передачу данных в двух направлениях поочередно. Модемы на каждом конце канала устанавливаются в состояние приема или передачи с помощью сигналов управления. *Дуплексный канал* позволяет передавать данные одновременно в двух направлениях. Это обеспечивается за счет

использования четырехпроводной линии связи (два провода служат для передачи, а два других – для приема данных), или двух полос частот.

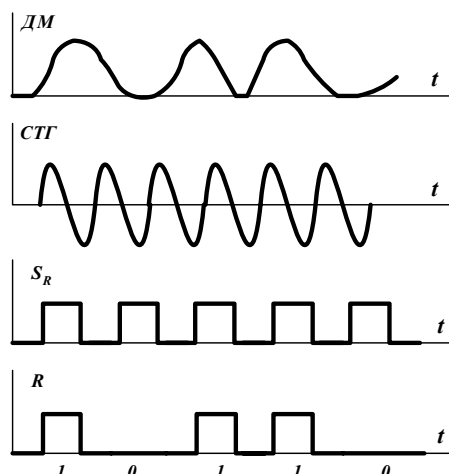


Рис. 4.8. Сигналы в демодуляторе

Таблица 4.1. Характеристики модемов ЕС ЭВМ

Обозначение	Тип канала	Режим работы канала	Пропускная способность основного канала, бит/с	Пропускная способность обратного канала, бит/с
Модем-200 ЕС-8001	ТфК, ТфНК	Д, ПД	200	–
Модем-1200 ЕС-8006	ГфК, ТфНК	Д, ПД	600, 1200	75
Модем-2400 ЕС-8011	ТфНК	Д, ПД	1200, 2400	75
Модем-4800 ЕС-8015	ТфНК	Д	2400, 4800	75
Модем-4800 ЕС-8019	Широкополосный	Д	2400, 4800	75

Примечание. ТфК и ТфНК – телефонный коммутируемый и некоммутируемый канал; Д и ПД – дуплексный и полудуплексный режим.

Для повышения достоверности передачи данных основной канал может снабжаться дополнительным вспомогательным каналом небольшой пропускной способности – обратным каналом. Например, при скорости передачи 1200 бит/с обратный канал работает со скоростью 75 бит/с. Такой канал создается выделением в полосе частот дополнительного канала с неширокой подполосой, используемой для передачи служебной информации в обратном направлении. По обратному каналу передаются сигналы, подтверждающие прием блоков данных. Если в принятом блоке обнаружена ошибка, то посылается сигнал на повторную передачу этого блока.

Характеристики типичных модемов, выпускаемых в рамках ЕС ЭВМ, приведены в табл. 4.1.

В состав АПД может включаться устройство защиты от ошибок (УЗО), обеспечивающее повышение достоверности данных путем обнаружения и исправления возникающих при передаче ошибок. Однако в настоящее время функции защиты от ошибок возлагаются, как правило, на окончное оборудование данных – ЭВМ, мультиплексоры передачи данных и программируемые абонентские пункты. Применяются УЗО в редких случаях – лишь при подключении к каналу связи терминального оборудования, не имеющего средств для логической обработки данных.

Для работы с коммутируемыми каналами телефонной и телеграфной сети на АПД возлагается функция установления соединения с абонентом. Эта функция может

выполняться ручным или автоматическим способом. При ручном способе вызова соединение в телефонной сети устанавливается с помощью телефонного аппарата (рис. 4.9, а). После этого переключатели «телефон – данные» устанавливаются в положение «данные», подключая к каналу модемы. Для того чтобы уменьшить время создания соединения, АПД снабжается *автоматическим вызывным устройством* (АВУ) (рис. 4.9, б), которое получает от оконечного оборудования данных (от ЭВМ) номер вызываемого абонента, преобразует номер в сигналы тональной частоты и принимает ответ вызываемой стороны. В процессе вызова абонента принимает участие ЭВМ, мультиплексор передачи данных, АВУ и абонентский пункт, на котором устанавливаемся автоответчик.

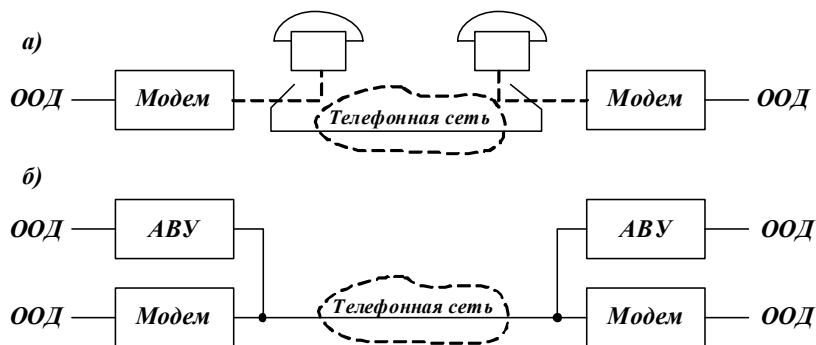


Рис. 4.9. Установление соединений через коммутируемый канал

Интерфейсы АПД. Для унификации технических средств телеобработки, в частности АПД, проведена стандартизация интерфейсов: АПД – линия (канал) связи и АПД – оконечное оборудование данных. Состав интерфейсов представлен на рис. 4.10. Интерфейс (стык) *C1* устанавливает логические и электрические аспекты сопряжения АПД с каналами связи и для телефонных каналов определяются отраслевым стандартом ОСТ4 ГО.208.004. Абонентский интерфейс *C2* (ГОСТ 18145-81) между АПД и ООД определяется рекомендацией Международного консультативного комитета по телеграфии и телефонии (МККТТ) V.24, которая устанавливает логические, электрические и конструктивные элементы сопряжения модемов и АВУ с ООД. Для сопряжения модемов с ООД используется 34 линии с номерами 101 – 134, а для сопряжения АВУ с ООД – 13 линий с номерами 201 – 213. Интерфейс *C3* определяет подключение УЗО к ООД и установлен стандартом ГОСТ 18146–72. Интерфейс *C3* отличается от интерфейса *C2* наличием цепей для параллельной передачи данных по пяти – восьми линиям и отсутствием цепей синхронизации и управления вызывными устройствами.

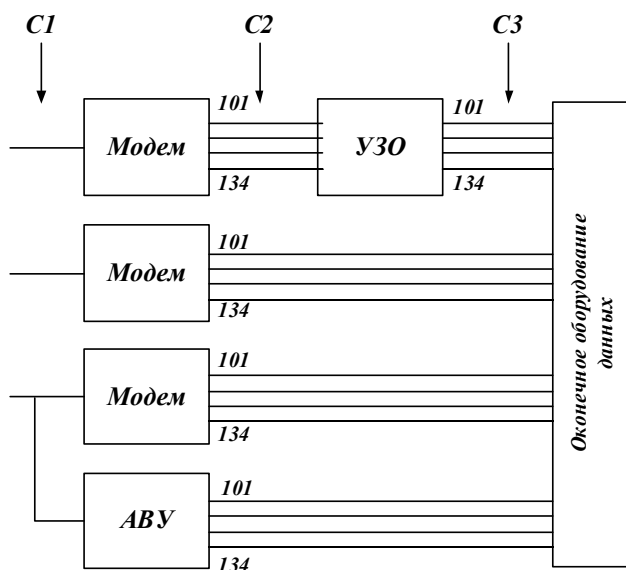


Рис. 4.10. Интерфейсы ЛПД

4.3. СОПРЯЖЕНИЕ ЭВМ С КАНАЛАМИ СВЯЗИ

Основная функция средств сопряжения ЭВМ с каналами связи – обмен данными между каналом ввода – вывода ЭВМ и аппаратурой передачи данных, т. е. каналами связи. Обмен данными программируется командами ввода – вывода, выполнение которых сводится к вводу в заданную область оперативной памяти ЭВМ сообщений, поступающих по каналу связи, и выводу сообщений, хранимых в оперативной памяти, в канал связи. Дополнительные функции средств сопряжения ЭВМ с каналами связи – повышение достоверности передаваемых данных путем проверки принимаемых данных и перезапроса сообщений, содержащих ошибки, формирование слов, характеризующих состояние средств сопряжения (включены-выключены, заняты-свободны и др.), и контроль работоспособности средств сопряжения и каналов связи. Для сопряжения ЭВМ с каналами связи применяется следующие устройства: линейные адаптеры, мультиплексоры передачи данных и связные процессоры.

Линейный адаптер – устройство, обеспечивающее сопряжение ЭВМ с одним каналом передачи данных. Линейный адаптер (ЛА) подключается с одной стороны к интерфейсу ввода – вывода ЭВМ (к каналу ввода – вывода) и с другой стороны к АПД, обслуживающей канал связи. Линейный адаптер реализует следующие функции: интерпретирует команды ввода – вывода в сигналы, управляющие работой АПД; преобразует последовательность слов данных, поступающих по каналу ввода – вывода ЭВМ, в последовательность битов, передаваемых через АПД по каналу связи и выполняет обратное преобразование при приеме данных по каналу связи; добавляет при передаче, распознает и устраняет при приеме служебные комбинации битов. В связи с тем, что в системах телеобработки к ЭВМ подключается значительное число каналов связи, использование для каждого канала линейного адаптера приводит к большим затратам оборудования. Поэтому линейные адаптеры не получили широкого распространения.

Мультиплексор передачи данных – устройство, обеспечивающее сопряжение ЭВМ с несколькими каналами связи. Структура МПД укрупненно представлена на рис. 4.11. Устройство сопрягает интерфейс ввода – вывода ЭВМ, например выход мультиплексного канала, с АПД, обслуживающей каналы связи. По интерфейсу ввода - вывода в МПД передаются команды, данные и посылаются в ЭВМ байты, характеризующие состояние МПД, линейных адаптеров и каналов связи. Блок сопряжения с каналом ввода-вывода (БСК) реализует интерфейсные функции, обеспечивая формирование сигналов взаимодействия с каналом через интерфейс ввода-вывода, прием и передачу байтов данных, составляющих передаваемые сообщения. Блок осуществляет обмен байтами данных с соответствующими адаптерами. Последние преобразуют выводимые байты в последовательность бит, передаваемых в АПД, а при вводе данных выполняют обратное преобразование.

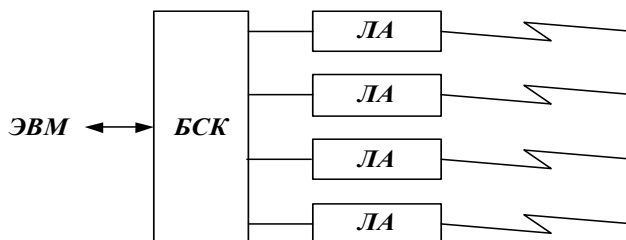


Рис. 4.11. Мультиплексор передачи данных

Для расширения функциональных возможностей МПД в его состав вводится микропроцессор, позволяющий проводить программную обработку принимаемых и отправляемых сообщений. Такие МПД называются программируемыми. За счет программируемости МПД существенно снижается нагрузка на центральный процессор, который освобождается от функций повышения достоверности передаваемых данных, установления соединений по коммутируемым каналам связи и индикации состояний МПД,

АПД и каналов связи. Микропроцессор содержит оперативную и перепрограммируемую постоянную память и включается между БСК и ЛА, так что адаптеры можно рассматривать в качестве периферийных устройств микропроцессора, Оперативная память микропроцессора используется в качестве буферной памяти, в которой хранятся сообщения, передаваемые между ЭВМ и каналами связи. За счет этого снижается частота обмена данными между ЭВМ и МПД, что позволяет подключать к МПД высокоскоростные каналы связи.

Для полного отделения предварительной обработки сообщений и управления средствами передачи данных от функций центральной ЭВМ используются связанные процессоры. Связной процессор состоит из микро- или мини-ЭВМ, к которой в качестве периферийных устройств подключают ЛА и АПД. Связной процессор реализует все функции управления каналами связи, а также предварительную обработку сообщений и обеспечивает повышение достоверности передаваемых данных. За счет этого существенно уменьшается нагрузка на центральную ЭВМ. Связной процессор подключается к каналу ввода – вывода ЭВМ через адаптер межмашинного обмена в качестве сателлитного процессора центральной ЭВМ.

Таблица 4.2. Характеристики МПД ЕС ЭВМ

Обозначение		Число каналов		Пропускная способность каналов, бит/с
		полудуплексных	дуплексных	
МПД-1А	ЕС-8400	15	7	50–4800
МПД-1	ЕС-8401	64	4	50–2400
МПД-2	ЕС-8402	176	88	50–4800
МПД-3	ЕС-8403	4	2	50–48000
МПД-4	ЕС-8404	12	–	200–1200
МПД-10	ЕС-8410	32	–	50–2400
УМПД	ЕС-8421	20	1	50–200 50–1200

Характеристики МПД ЕС ЭВМ представлены в табл. 4.2. Устройства могут комплектоваться различной АПД (модемами): МПД-2 и МПД-4 являются программируемыми. Удаленный мультиплексор (УМПД) служит для сопряжения нескольких низкоскоростных каналов, используемых абонентами, с высокоскоростным каналом, ведущим к ЭВМ. Удаленный МПД ЕС-8421 обеспечивает передачу сообщений 20 абонентов, связанных с ним через низкоскоростные каналы (50–200 бит/с), по дуплексному среднескоростному (1200 бит/с) каналу.

4.4. АБОНЕНТСКИЕ ПУНКТЫ

Абонентские пункты – терминалы системы телеобработки, предназначенные для обмена данными между пользователями и ЭВМ через каналы связи. Состав АП представлен на рис. 4.12. Оконечными устройствами АП являются, с одной стороны, устройства ввода – вывода – дисплеи, ввод с перфолент, перфокарт и магнитных лент, печать и вывод на магнитную ленту, а с другой стороны, аппаратура передачи данных (модем) и при работе с коммутируемым каналом связи – вызывное устройство. Устройства ввода – вывода и АПД сопрягаются устройством обмена данными, которое обеспечивает передачу данных между АПД и каждым УВВ, а также, возможно, обмен данными между УВВ. Обмен данными сводится к формированию из битов или символов, поступающих от АПД или УВВ, блоков данных и выводу блоков в виде последовательности битов или символов на заданное устройство – АПД или УВВ. Устройство обмена данными обеспечивает согласование скоростей работы АПД и различных УВВ за счет буферизации символов или блоков данных, которые принимаются в память с одной скоростью и выводятся из нее с другой скоростью.

Управление устройствами АП осуществляется устройством управления. Для включения АП, индикации и управления режимами работы служит пульт АП.

Функции в АП могут быть реализованы аппаратурно (жесткая логика) или с помощью программ. При аппаратурной реализации состав функций жестко зафиксирован и не может изменяться в процессе эксплуатации АП. Программируемые АП строятся на базе микро- или мини-ЭВМ, к которой подключаются АПД и УВВ. Машина управляет работой АПД и УВВ, контролирует состояние канала связи и реализует требуемые алгоритмы взаимодействия с системой телеобработки, повышения достоверности данных, формирования сообщения и обмена данными между УВВ. Гибкость программируемых АП и возможность реализации в них широкой номенклатуры функций взаимодействия с системой телеобработки и с пользователями приводит ко все большему распространению этих АП в системах телеобработки. Программируемые АП можно рассматривать в качестве интеллектуальных терминалов в системах телеобработки, позволяющих наряду с традиционными функциями накапливать и редактировать данные и даже производить их первичную обработку. Характеристики типичных АП ЕС ЭВМ представлены в табл. 4.3. Устройства АП-6, АП-31, АП-32 и АП-50 являются программируемыми.

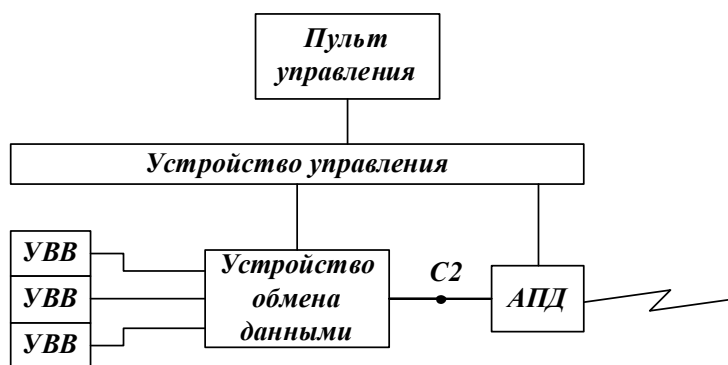


Рис. 4.12. Абонентский пункт

Таблица 4.3. Характеристики абонентских пунктов ЕС ЭВМ

Обозначение	Тип канала	Режим работы канала	Пропускная способность канала, бит/с	Устройство ввода	Устройство вывода
АП-1 ЕС-8501	ТфК2, ТгНК	ПД	200, 1200	Кл, ПфЛ, ККП	ПфЛ, ККП, ПМ
АП-2 ЕС-8502	ТфК2	ПД	200	То же	То же
АП-3 ЕС-8503	ТфНК2	ПД	600, 1200	«	«
АП-4 ЕС-8504	ТфНК4	Д	1200, 2400	Кл, ПфЛ, ПфК, НМЛ	ПфЛ, ПМ, НМЛ, АЦПУ
АП-5 ЕС-8505	ТфНК2	Д, ПД	200, 600, 1200	Кл, ПфЛ, ПфК	ПфЛ, АЦПУ
АП-11 ЕС-8511	ТфНК4	Д	1200, 2400	ПфЛ, ПфК	ПфЛ, ПфК
АП-50 ЕС-8550	ТфНК4	Д	600–9600	Кл, ПфЛ, ПфК, НМЛ, НМД	Т, ПМ, НМЛ, НМД
АП-61 ЕС-8561	ТфНК2, ТфНК4	ПД	200, 1200, 2400	Кл	PM, Т
АП-63 ЕС-8563	ТфНК4	Д	2400	Кл	Т, ПМ
АП-70 ЕС-8570	ТфК2, ТгНК	ПД	100	Кл	ПМ

Примечание. ТфК2 – телефонный коммутируемый или некоммутируемый 2-проводной канал, ТфНК2 и ТфНК4 – телефонный некоммутируемый 2- и 4-проводной канал; ТгНК – телеграфный некоммутируемый канал; Д и ПД – дуплексные и полудуплексный режим; Кл – клавиатура; ПфЛ – перфолента; ПфК – перфокарта; ККП – карты с краевой перфорацией; ПМ – пишущая машинка; АЦПУ – алфавитно-цифровое (строчное) печатающее устройство; Т – терминал (дисплей)!

4.5. ПРОГРАММНЫЕ СРЕДСТВА

Телеобработка данных состоит в приеме сообщений, посылаемых пользователями через абонентские пункты, обработке принятых сообщений с помощью системных обрабатывающих или прикладных программ и передаче формируемых в ЭВМ сообщений заданным абонентским пунктам. Иногда при телеобработке необходимо выполнять ряд дополнительных функций: установление соединений с абонентами, связанными с ЭВМ через коммутируемые каналы; активизацию и прекращение работы абонентов; редактирование принимаемых и передаваемых сообщений, например вставку даты, времени суток и порядковых номеров сообщений; обработку сообщений, содержащих ошибки; повторную обработку (рестарт) после отказов технических средств. Для реализации указанных функций необходимы программы управления вводом – выводом данных через МПД и АПД. Телеобработка поддерживается программными средствами, включаемыми в состав операционной системы.

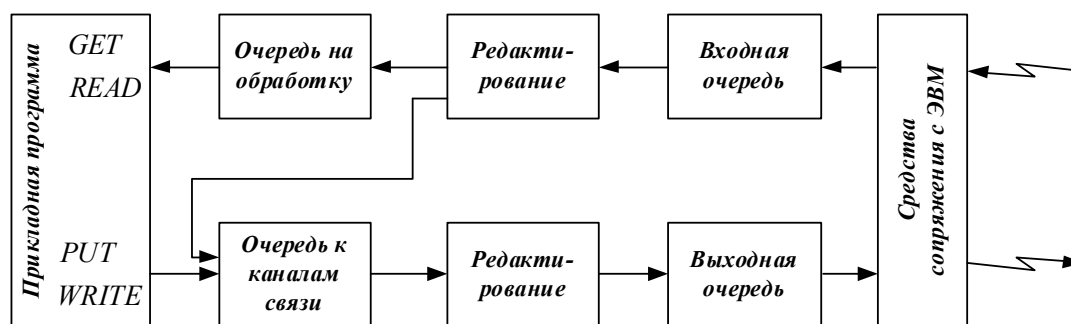


Рис. 4.13. Обработка сообщений

Телеобработка данных организуется по схеме, представленной на рис. 4.13. Сообщения, поступающие в ЭВМ по каналам связи, накапливаются во входной очереди, затем редактируются с помощью средств телеобработки и поступают в очередь к программам обработки. Прикладная программа обращается к очереди сообщений как к набору данных с последовательной организацией и по макрокомандам GET, READ выбирает сообщения для обработки. При этом процесс приема сообщений и процесс обработки протекают асинхронно. Формируемые прикладной программой сообщения выводятся в очередь к каналам связи с помощью макрокоманд PUT, WRITE. Затем сообщения могут вновь редактироваться, после чего поступают в выходную очередь, откуда выводятся в порядке поступления в каналы связи и по ним достигают пользователей. Программные средства телеобработки создают возможность обмена сообщениями между пользователями. В таком случае сообщение, минуя прикладные программы, направляется в очередь к каналу связи и после редактирования выводится по каналу к адресуемому абоненту.

Описанная схема и взаимодействие прикладных программ с потоками вводимых – выводимых сообщений обеспечивается программами телекоммуникационного метода доступа (ТМД). В операционной системе ЕС ЭВМ используются два метода доступа к данным: базисный (БТМД) и общий (ОТМД). Базисный метод предоставляет пользователям простейшие средства работы с сообщениями, переливаемыми по каналам связи. Общий метод значительно расширяет возможности базисного метода, обеспечивая автоматическое управление ресурсами системы телеобработки и предоставляя пользователям гибкий язык высокого уровня для управления потоками сообщений между абонентскими пунктами

(коммутации сообщений) и между абонентскими пунктами и прикладными программами (обработка сообщений). С помощью языка управления сообщениями можно активизировать каналы связи и управлять передачей сообщений между ЭВМ и абонентскими пунктами, вставлять и удалять символы управления каналами связи, получать, использовать и освобождать буфера для хранения сообщений после их приема и перед отправлением, направлять сообщения к абонентским пунктам и прикладным программам, проводить корректировку и специальную обработку сообщений, содержащих ошибки. Общий метод позволяет программисту обрабатывать сообщения так же просто, как и при использовании традиционных устройств ввода – вывода ЭВМ.

Порядок телеобработки под управлением программных средств ОТМД следующий:

1. Сообщение подготавливается на абонентском пункте – набирается на клавиатуре или вводится с перфорированной или магнитной ленты.
2. Абонентский пункт передаст сообщение по каналу связи через мультиплексор передачи данных и мультиплексный канал в основную память ЭВМ.
3. Сообщение вводится во входную очередь, проходит первоначальный контроль и при необходимости перекодируется во внутренний код ЭВМ. При обнаружении ошибки в передающий пункт направляется сигнал об ошибке, инициирующий повторную передачу сообщения.
4. Сообщение может быть подвергнуто редактированию – удалению или вставке управляющих символов, включению даты, времени и порядкового номера сообщения.
5. После входной обработки сообщение поступает в очередь назначения – либо к прикладной программе, либо к пункту назначения (каналу связи).
6. Когда прикладная программа выдает макрокоманды GET или READ, средства ОТМД передают данные из сообщения в рабочую область прикладной программы. После этого сообщение исключается из очереди.
7. Прикладная программа генерирует ответные сообщения, которые выводятся из нее по макрокоманде PUT или WRITE и поступают в выходную очередь. При этом к данным добавляется заголовок и прочие атрибуты сообщения.
8. Выходные сообщения могут подвергаться различного рода обработке (преобразование, в код передачи для пункта назначения, редактирование, фиксация в системном журнале и подсчет числа выводимых сообщений).
9. Средства ОТМД выбирают сообщение из выходной очереди (по порядку) и передают его через соответствующий канал связи в абонентский пункт.

При использовании ОТМД прикладные программы практически не зависят от специфики технических средств телеобработки – каналов связи и аппаратуры сопряжения ЭВМ с каналами.

ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

5.1. СТРУКТУРА И ХАРАКТЕРИСТИКИ.

Для создания крупномасштабных систем обработки данных ЭВМ и вычислительные комплексы, обслуживающие отдельные предприятия и организации, объединяются с помощью средств передачи данных в вычислительные сети, обеспечивающие отрасли производства и регионы,

Структура. Структура вычислительной сети представлена на рис. 5.1. Вычислительная сеть разделяется на три взаимосвязанные подсети: базовую сеть передачи данных (СПД), сеть ЭВМ и терминальную сеть.

Базовая СПД – совокупность средств для передачи данных между ЭВМ. Сеть передачи данных состоит из линий связи и узлов связи. Узел связи – совокупность средств коммутации и передачи данных в одном пункте. Узел связи принимает данные, поступающие по каналам связи, и передает данные в каналы, ведущие к абонентам. Узел связи реализуется на основе коммутационной ЭВМ и аппаратуры передачи данных. *Коммутационная ЭВМ* управляет приемом и передачей данных и, в частности, выбирает целесообразный путь передачи данных. Базовая СПД является ядром вычислительной сети, обеспечивающим физическое объединение ЭВМ и прочих устройств.

Сеть ЭВМ – совокупность ЭВМ, объединенных сетью передачи данных. Сеть ЭВМ включает в себя главные и терминальные ЭВМ. Главная ЭВМ (ГВМ) выполняет задания абонентов сети – пользователей. *Терминальные ЭВМ* (ТВМ) предназначены для сопряжения терминалов с базовой СПД. Основная функция сопряжений сводится к преобразованию данных и форму, обеспечивающую их передачу средствами базовой сети и вывод данных на терминалы.

Терминальная сеть – совокупность терминалов в терминальной сети передачи данных. *Терминалы* – устройства, с помощью которых абоненты осуществляют ввод и вывод данных. В терминальной сети могут использоваться интеллектуальные терминалы и абонентские пункты. В состав интеллектуального терминала входит процессор, обеспечивающий локальную обработку данных – редактирование текстов, отображение данных в специальной форме, хранение данных и манипуляции с ними и т. д. Абонентский пункт состоит из взаимосвязанных устройств ввода – вывода, обеспечивающих ввод данных от нескольких источников и вывод данных в различной форме – на экраны дисплеев, печатающие устройства, устройства вывода графической информации и др. Для подключения терминалов к ЭВМ используются линии связи и обслуживающие их удаленные мультиплексоры передачи данных, в совокупности образующие *терминальную сеть передачи данных*.

Контроль состояния вычислительной сети и управление ее функционированием обеспечивается *административной системой*, включающей в себя ЭВМ, терминальное оборудование и программные средства, с помощью которых производится включение и выключение сети и ее компонентов, контролируется работоспособность сети, устанавливаем режим функционирования компонентов, систем и сети в целом, учитывается объем услуг, предоставляемых абонентам сетью, и т. д.

Отдельные вычислительные сети могут быть связаны между собой с помощью линий связи, подключаемых к узлам межсетевой связи. В узле межсетевой связи используется

ЭВМ, обеспечивающая согласование и преобразование данных при передаче их между двумя телями.

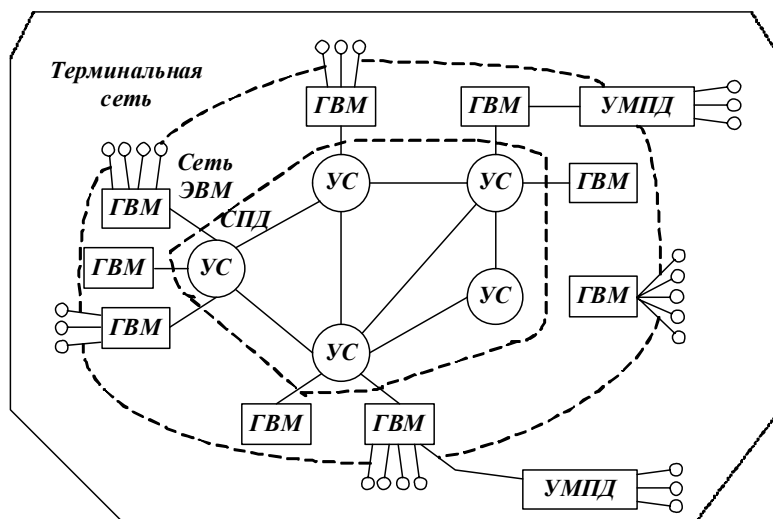


Рис 5.1. Структура вычислительной сети

Эффект сетевой обработки данных. Основной эффект от объединения ЭВМ и терминалов в вычислительную сеть – это полная доступность ресурсов сети для пользователей. Пользователи, подключенные к сети, имеют доступ ко всем главным ЭВМ, входящим в сеть, и, следовательно, имеют возможность использовать память этих ЭВМ для хранения данных и процессоры для их обработки. Пользователям доступно программное обеспечение, имеющееся в сети, и базы данных, размещенные в ЭВМ сети, что позволяет оперативно использовать программы и базы данных. Как правило, сети предоставляют возможность параллельной обработки данных многими ЭВМ. Возможно построение распределенных баз данных, размещенных в памяти многих ЭВМ, а за счет этого – создание сложных информационных структур. Информационные связи между пользователями позволяют коллективам пользователей решать задачи моделирования сложных систем, выполнять проектные и другие работы, опирающиеся на распределенные между многими ЭВМ программное обеспечение и базы данных. Таким образом, сетевая обработка данных – качественно новая организация обработки: в значительной степени увеличивается предел сложности и скорость решения задач, требующих участия больших коллективов работников.

Вычислительные сети позволяют повысить уровень загрузки ЭВМ, программного обеспечения и баз данных. Это обусловлено двумя факторами. Во-первых, вычислительная сеть обслуживает большое число пользователей, поэтому нагрузка, создаваемая всеми пользователями, в меньшей степени подвержена колебаниям, чем нагрузка, создаваемая отдельным пользователем или группой. Этот эффект имеет статистическую природу и оценивается дисперсией среднего значения нагрузки, создаваемой пользователями. Так, если среднее квадратическое отклонение нагрузки, создаваемой одним пользователем, равно a , то и пользователей создают суммарную нагрузку, среднее квадратическое отклонение которой равно σ/\sqrt{n} , т. е. колебания нагрузки, создаваемой, например, 100 пользователями, в 10 раз меньше, чем у создаваемой одним пользователем. Следовательно, увеличивается вероятность того, что в каждый момент времени существует работа для каждого компонента сети, т. е. увеличивается загрузка ресурсов сети. Второй фактор, позволяющий повысить уровень загрузки, – стабилизация нагрузки на сеть, когда сеть охватывает территорию, расположенную в нескольких часовых поясах. Эффект стабилизации особенно существен для эксплуатации специализированных и проблемно-ориентированных ЭВМ (ЭВМ с матричными процессорами), аналого-цифровых вычислительных комплексов, информационно-справочных систем и др.

Как показывает практика, за счет расширения возможностей обработки данных и лучшей загрузки ресурсов стоимость обработки данных средствами сети снижается в полтора раза и более, по сравнению с обработкой данных на несвязанных ЭВМ.

Характеристики. Основные характеристики вычислительной сети – операционные возможности, время доставки сообщения, производительность и стоимость обработки данных.

Операционные возможности сети – перечень основных действий по обработке данных. Главные ЭВМ, входящие в состав сети, обеспечивают пользователей всеми традиционными видами обслуживания, средствами автоматизации программирования, доступом к пакетам прикладных программ, базами данных и т. д. Наряду с этим вычислительная сеть может предоставлять пользователям следующие дополнительные виды услуг:

- 1) удаленный ввод заданий – выполнение заданий, поступающих с любых терминалов, на любой главной ЭВМ в пакетном или диалоговом режиме;
- 2) передачу файлов (наборов данных) между ЭВМ сети;
- 3) доступ к удаленным файлам – обработку файлов, хранимых в удаленных ЭВМ;
- 4) защиту данных и ресурсов от несанкционированного доступа;
- 5) передачу текстовых и, возможно, речевых сообщений между терминалами (пользователями);
- 6) выдачу справок об информационных и программных ресурсах;
- 7) распределенные базы данных, размещаемые в нескольких ЭВМ;
- 8) распределенную обработку – параллельное выполнение задачи несколькими ЭВМ.

Как минимум в сетях реализуются первые шесть видов дополнительных услуг. Работа с распределенными базами данных и распределенная обработка обеспечиваются только в наиболее развитых вычислительных сетях.

Производительность сети представляет собой суммарную производительность главных ЭВМ. При этом обычно производительность главных ЭВМ означает номинальную производительность их процессоров.

Время доставки сообщений определяется как статистическое среднее времени от момента передачи сообщения в сеть до момента получения сообщения адресатом.

Цена обработки данных формируется с учетом стоимости средств, используемых для ввода – вывода, передачи, хранения и обработки данных. На основе цен рассчитывается стоимость обработки данных, которая зависит от объема используемых ресурсов вычислительной сети (количество передаваемых данных, процессорное время), а также режима передачи и обработки данных.

Указанные характеристики зависят от структурной и функциональной организации сети, т. е. от набора параметров, основные из которых: структура вычислительной сети (состав ЭВМ, структура базовой СВД и терминальной сети), метод передачи данных в базовой сети, способы установления соединений между взаимодействующими абонентами, выбора маршрутов передачи данных и т. д. Кроме того, характеристики сети зависят от нагрузки, создаваемой пользователями. Нагрузка определяется числом активных терминалов (пользователей) и интенсивностью взаимодействий.

пусковой способности канала и, возможно, скорости формирования и анализа специальных кодовых последовательностей, характеризующих состояние канала. Протокол *УФК* вводит интерфейс, устанавливающий стандартный для всех систем порядок взаимодействия с каналом.

На рисунке уровень 2 разделен на два подуровня: *2.1 – управление доступом к каналу*; *2.2 – управление информационным каналом*. Порядок функционирования этих подуровней регламентируется одноименными протоколами *УДК* и *УИК*. Протокол *УДК* устанавливает процедуру передачи данных через канал, коллективно используемый системами, и процедуру селекции данных, передаваемых по каналу. Протокол *УИК* устанавливает порядок обеспечения достоверности данных при передаче через физический канал, подверженный воздействию помех: формируются проверочные коды при передаче данных, а также в пункте приема (проверка корректности принимаемых данных), и при обнаружении искажения передача данных повторяется. В большинстве ЛВС отсутствует необходимость в сетевом уровне управления, поэтому на рисунке этот уровень не выделен. Необходимость в этом уровне появляется при комплексировании нескольких ЛВС, содержащих моноканалы. Однако и в этом случае функции сетевого уровня оказываются, как правило, достаточно простыми.

На транспортном уровне протокол *управления передачей (УП)* обеспечивает единый транспортный интерфейс для процессов высокого уровня. Транспортный уровень ликвидирует различия между потребностями процессов в обмене данными и ограниченными возможностями информационного канала, организуемого нижними уровнями управления. За счет этого при программировании прикладных процессов нет необходимости учитывать специфику функционирования моноканала.

Протоколы высокого уровня – *управления сеансами (УС)*, *представлением данных (УПД)* и *прикладными процессами (УПП)* – по своим функциям аналогичны соответствующим протоколам глобальных сетей. В ЛВС на представительском уровне реализуется доступ терминалов к процессам, программ к удаленным файлам, передача файлов, удаленный ввод заданий, обмен графической информацией и др.

При использовании моноканала сложность процедур управления на физическом, канальном и транспортном уровне невелика. Поэтому эти уровни управления удается эффективно реализовать в основном техническими средствами – специальным контроллером, называемым *сетевым адаптером (станцией)* ЛВС. Адаптер сопрягает внутреннюю магистраль или интерфейс ввода – вывода ЭВМ или другой системы сети с последовательным интерфейсом. По существу адаптер в совокупности с физическим каналом образует информационный моноканал (рис. 6.3), к которому подключаются системы сети, выступающие в этом случае в качестве абонентов моноканала. Сетевые адаптеры реализуют, во-первых, сетевые функции – управление физическим каналом, информационным каналом и передачей и, во-вторых, интерфейсные функции – подключение адаптера к интерфейсу системы, как правило, параллельному.

Локальные вычислительные сети используются для комплексирования в основном микро- и мини-ЭВМ. На рис. 6.4, *а* представлена типичная конфигурация ЛВС, предназначенной для расширения возможностей персональных ЭВМ (*ПЭВМ*). Здесь используются простейшие персональные ЭВМ, состоящие из микро-ЭВМ и дисплея. Вычислительные возможности персональных ЭВМ расширяются путем подключения их к *НМД* и системным устройствам вывода – *АЦПУ* и графопостроителям (*ГП*) используемым в режиме коллективного доступа. За счет этого, каждый пользователь имеет возможность работать с большими наборами данных, размещаемыми вне персональных ЭВМ, и высокоскоростными устройствами вывода.

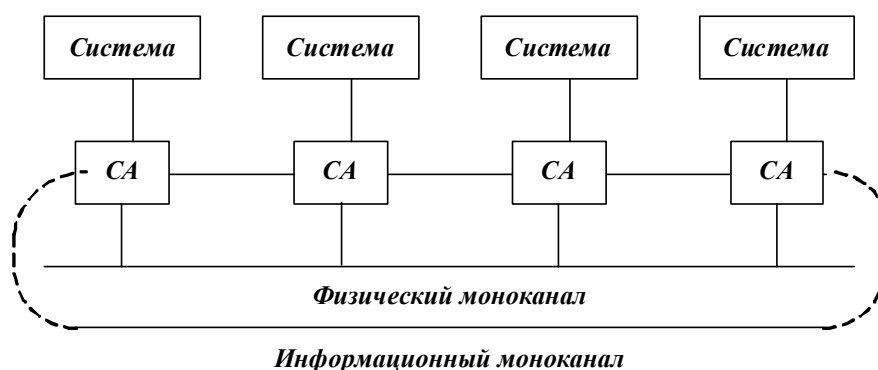


Рис. 6.3. Организация моноканала

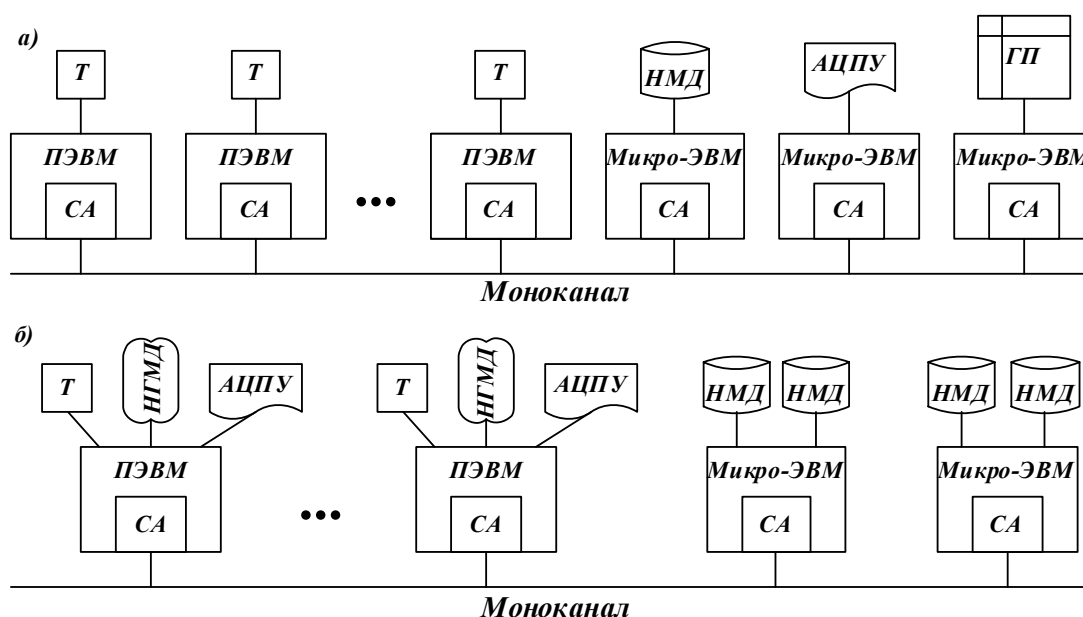


Рис. 6.4. Состав ЛВС

На рис. 6.4, б представлена более мощная сеть, состоящая из персональных ЭВМ, оснащенных дисплеем, накопителем на гибких магнитных дисках (НГМД) и печатающим устройством, и мини-ЭВМ предоставляющих пользователям значительные вычислительные возможности и базы данных, размещаемые в НМД. Стоимость средств подключения к сети (сетового программного обеспечения систем и адаптеров) оказывается значительно ниже стоимости ресурсов, предоставляемых пользователям от сети. Этим обусловлена эффективность ЛВС. Кроме того, объединение ЭВМ в сеть позволяет решать более сложные задачи и создает качественно новые условия для параллельной обработки, данных коллективами пользователей ЛВС. Отказы в отдельных системах не приводят к отказу сети в целом. Поэтому ЛВС обладают более высокой надежностью, чем системы обработки данных такой же мощности, построенные на основе единственной ЭВМ с подключенным к ней терминальным оборудованием.

Область применения ЛВС чрезвычайно обширна: системы автоматизации проектирования и технологической подготовки производства, управления производством и технологическими комплексами, конторские системы, бортовые системы управления и др. Системы автоматизации проектирования и технологической подготовки производства, а также конторские (системы управления снабжением, сбытом, складами, распределением транспорта, системы перечисления платежей и др.) обеспечивают высокую информативность

за счет концентрации данных в едином комплексе и оперативного доступа к данным и средствам их обработки.

Локальные сети являются эффективным способом построения сложных систем управления производственными участками, цехами и предприятиями. В таких системах для управления станками с ЧПУ, промышленными роботами, автоматическими транспортными и складскими средствами используются, как правило, микро-ЭВМ. Системы управления производственными подразделениями связываются с системами автоматизации проектирования, технологической подготовки производства и административного управления производством, образуя интегрированные производственные комплексы, решающие всю совокупность задач подготовки высокоавтоматизированного производства и управления им. В бортовых системах управления использование моноканала для сопряжения датчиков, устройств отображения и ЭВМ, решающих локальные задачи управления и контроля, позволяет значительно уменьшить число соединений и координировать работу многих подсистем, в результате чего снижается стоимость системы управления и повышается качество управления судами, самолетами и другими объектами.

6.2. МОНОКАНАЛЫ

Моноканал ЛВС состоит из канала передачи данных и сетевых адаптеров, сопрягающих ЭВМ с каналом (см. рис. 6.3).

Каналы передачи данных. Канал передачи данных состоит из кабеля, по которому передаются сигналы, являющиеся носителями двоичных значений 0 и 1, последовательно передаваемых по каналу,

Таблица 6.1. Характеристики кабелей

Тип кабеля	Скорость передачи данных, Мбит/с, на расстояние			Стоимость (в условных единицах)	Помехоустойчивость
	180 м	600 м	1900 м		
Экранированные пары: с индивидуальной экранировкой пары	1	0,3	0,1	1-6	отличная
с общей экранировкой пар	2	0,35	0,15	1-3	удовлетворительная
Экранированные пары, 100 Ом, с индивидуальной экранировкой пары	3,5	0,5	0,15	3-11	отличная
Коаксиальный кабель: 75 Ом; 6,35 мм	7	1,8	0,6	0,4	отличная
75 Ом; 12,7 мм	12	2,5	1,0	0,7	отличная
93 Ом	15	5,0	0,8	0,8	хорошая
Волоконно-оптическая линия: с плавным профилем показателя преломления и полосой пропускания					
200 МГц	2056	518	207	2,1	отличная
600 МГц	6027	1512	615	3,5	отличная
со ступенчатым профилем показателя преломления	196	49	20	2,1	отличная

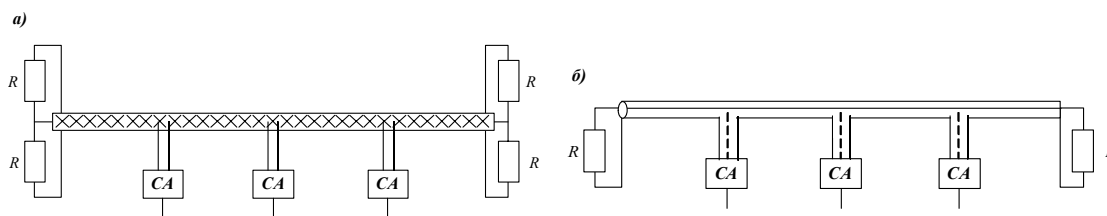


Рис.. 6.5. Подключение адаптеров к магистральному каналу

Для связи между системами ЛВС используются три типа кабелей: экранированные пары проводов, коаксиальные кабели и волоконно-оптические линии (табл. 6.1). Как видно из таблицы, пропускная способность кабеля уменьшается примерно пропорционально его длине, т. е. с увеличением длины кабеля в 10 раз пропускная способность уменьшается примерно во столько же раз. Помехоустойчивость канала также уменьшается с увеличением его длины. При отсутствии источников сильных электромагнитных помех уровень помехоустойчивости пар и коаксиальных кабелей при длине 10^2 м не ниже $10^{-8} - 10^{-9}$ искажений на бит передаваемых данных. Волоконно-оптические линии нечувствительны к электромагнитным помехам и не являются источниками электромагнитного излучения. Экранированные пары применяются в каналах с пропускной способностью до 1 Мбит/с, а коаксиальные кабели – в каналах с пропускной способностью 1–10 Мбит/с. Волоконно-оптические линии используются для создания каналов со сверхвысокой пропускной способностью– 10– 100 Мбит/с, а также в бортовых и производственных системах, работающих в условиях сильных электромагнитных помех.

Схемы подключения сетевых адаптеров к каналу на основе экранированных пар и коаксиального кабеля приведены на рис. 6.5, а и б соответственно. На концах сегментов витой пары и коаксиального кабеля устанавливаются схемы согласования, состоящие из резисторов. Подключение адаптеров производится через механический разъем, от которого отводится к адаптеру сегмент кабеля ограниченной длины – обычно до 10–20 м.

Волоконно-оптический канал (рис. 6.6) строится из сегментов, соединяющих соседние адаптеры (системы). Для передачи данных в каждом направлении используется самостоятельная линия. Сегмент линии состоит из передатчика *ПД*, формирующего электрические сигналы с требуемыми временными и электрическими параметрами, светодиода *Сд*, преобразующего электрические сигналы в световые, волоконно-оптической линии, по которой передаются световые сигналы, фотодиода *Фд*; воспринимающего световые сигналы и преобразующего их в электрические, и приемника *Пр*, формирующего сигналы с заданными временными и электрическими параметрами. Адаптеры подключаются к приемникам и передатчикам и транслируют электрические сигналы в следующие сегменты волоконно-оптического канала. Волоконно-оптические линии сопрягаются с приемопередатчиками, фото- и светодиодами через оптические разъемы *ОР*.

К сигналам, используемым для передачи данных по каналу предъявляются следующие основные требования: помехозащищенность, обеспечение синхронизации приема и передачи данных, максимальной пропускной способности канала и минимальных затрат оборудования в передатчиках, приемниках в канале. Помехозащищенность сигнала проявляется в возможности выделения данных, переносимых сигналом, при наличии помех. Синхронизация обеспечивает различимость битов, передаваемых по каналу. Это означает, что каждый бит представляется сигналом в таком виде, что возможно лишь однозначное его восприятие приемниками, т. е. сигнал, относящийся к одному биту, не может быть пропущен или интерпретирован в виде двух и более бит данных. Для увеличения пропускной способности и уменьшения затрат оборудования стремятся использовать сигналы минимальной длительности и по возможности более простой формы.

На рис. 6.7 представлены примеры сигналов, используемых для передачи данных в каналах ЛВС. Синхронизация процессов приема и передачи данных наиболее просто реализуется при использовании в моноканале четырехпроводной линии связи. Одна пара проводов служит для

передачи синхросигналов, следующих с периодом τ , а другая – для передачи единичных значений сигналов (рис. 6.7, а). Наличие на одной линии сигнала 1 при отсутствии сигнала на другой воспринимается приемником как 0. Наличие на каждой линия сигнала 1 воспринимается как 1. Другой порядок использования, линий в моноканале иллюстрируется рис. 6.7, б. Здесь одна линия используется для передачи сигналов 1, а другая – для передачи сигналов 0. Наличие в одном такте τ двух сигналов 1 свидетельствует об ошибке в моноканале. В четырехпроводных моноканалах, как правило, используются в качестве передающей среды витые пары с общим или разделенным экраном.

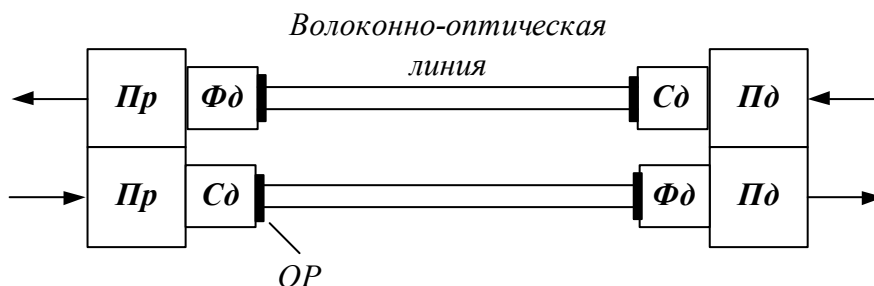


Рис. 6.6. Волоконно-оптический канал.

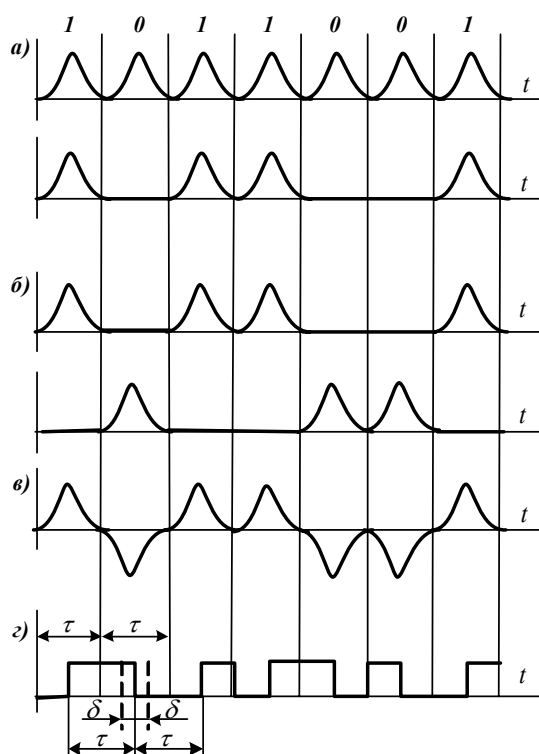


Рис. 6.7. Сигналы в каналах ЛВС

Для уменьшения затрат оборудования в передающей среде каналы строятся по однолинейной схеме – на основе одной витой пары, коаксиального кабеля или волоконно-оптической линии. При использовании электрических сигналов передача единичных и нулевых значений битов может производиться разнополярными сигналами. На рис. 6.7, в значение 1 передается импульсом положительной полярности, а значение 0 – импульсом отрицательной полярности. При этом биты данных представляются отдельными сигналами, которые синхронизируют процесс передачи данных. Для передачи данных по однолинейному моноканалу однополярными сигналами используется самосинхронизирующийся, так называемый манчестерский код, идеализированная форма сигналов

которого изображена на рис. 6.7, г. Значение 1 представляется переключением сигнала, например из 0 в 1, и тогда значение 0 – переключением сигнала из 1 в 0. Момент переключения сигнала отмечает середину такта, а направление переключения определяет значение передаваемого бита. В момент первого переключения сигнала приемник запускает систему синхронизации, которая разрешает прием сигнала в промежутке времени $[\tau - \delta, \tau + \delta]$, $\delta < \tau/2$, следующем за моментом поступления предыдущего сигнала. Сигнал, поступивший в этом промежутке времени, несет значение бита 0 или 1 и одновременно используется как очередной сигнал синхронизации. На основе его выделяется следующий промежуток времени $[\tau - \delta, \tau + \delta]$, в течение которого приемник готов принять очередной бит данных.

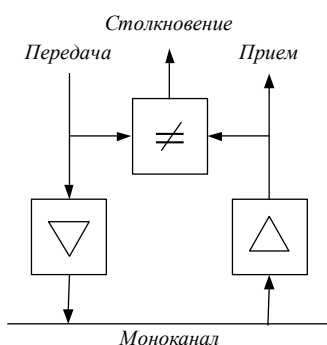


Рис. 6.8. Формирование сигнала о столкновении

Организация моноканалов. Наиболее существенные аспекты организации моноканала: 1) конфигурация ЛВС; 2) способ доступа к каналу; 3) протоколы управления физическим и информационным каналом. Эффективность организации моноканала характеризуется следующими основными показателями: 1) затратами оборудования в адаптере, зависящими от конфигурации ЛВС и протоколов канального уровня; 2) пропускной способности; 3) средней задержкой передачи данных; 4) надежностью, связанной в первую очередь с сохранением работоспособности моноканала при отказах отдельных адаптеров и систем сети. Значимость показателей эффективности зависит от области применения сети: в сетях персональных ЭВМ наиболее значимым может быть показатель затрат оборудования, а в производственных и бортовых системах – надежность моноканала.

Моноканалы ЛВС строятся из соображений надежности в основном по принципу распределенного управления доступом к каналу, поскольку при централизованном управлении выход из строя контроллера канала является катастрофическим для сети в целом. При распределенном управлении все станции, подключенные к каналу, функционируют одинаковым образом, получая информацию о занятости и освобождении канала только исходя из состояния физического канала. В этом случае для каждой станции канал является равнодоступным средством передачи данных, порядок доступа к которому определяется соответствующим протоколом. Прием данных производится путем *селекции* – выделения из множества данных тех, которые адресованы конкретной системе.

Различают три основных способа доступа к моноканалу: свободный, управляемый и комбинированный доступ. При *свободном (случайном) доступе* каждая система захватывает канал для передачи данных в произвольный момент времени. Если две или более системы одновременно передают данные в канал, за счет интерференции сигналов данные искажаются и подлежат повторной передаче, момент которой назначается по специальному, алгоритму. *Управляемый (детерминированный) доступ* основан на поочередном предоставлении системам разрешения на передачу данных. *Комбинированный доступ* основан на использовании свободного и управляемого доступа к каналу на разных фазах

работы систем. В ЛВС с магистральной структурой наиболее широко используется свободный, а с кольцевой структурой – управляемый доступ.

Свободный доступ с проверкой столкновений (СДПС). Этот способ наиболее широко применяется в магистральных структурах. Система захватывает канал и начинает передачу в любой момент времени. Поскольку централизованное управление отсутствует, две и более системы могут вести передачу пакетов одновременно. В этом случае происходит *столкновение* – интерференция пакетов, передаваемых одновременно, в результате чего все передаваемые пакеты искажаются. Столкновения обнаруживаются путем приема каждой системой передаваемого ею пакета (рис. 6.8). При этом биты, передаваемые в канал, сравниваются с битами, принимаемыми из канала. Если регистрируется несовпадение переданного и принятого бита, это свидетельствует о столкновении пакетов в канале. При обнаружении столкновения система прекращает передачу пакета и повторяет передачу через некоторое время. Для того чтобы уменьшить вероятность повторных столкновений, каждая система начинает повторную передачу через случайный промежуток времени с достаточно большим средним значением τ . Задержка передачи формируется как случайная величина, равномерно распределенная в интервале $[0, \tau_{\max}]$, где τ_{\max} – максимальная задержка повторной передачи.

Из-за столкновений реальная пропускная способность моноканала оказывается меньше номинальной пропускной способности физического канала. В целях упрощения математических выражений пропускные способности оценивают числом пакетов, передаваемых за время T , достаточное для передачи одного пакета по физическому каналу. Время передачи одного пакета по физическому каналу T называют *окном*,. Окно $T=L/V$, где L – длина пакета (точнее, кадра), бит, и V – пропускная способность физического канала, бит/с. С учетом сказанного пропускная способность моноканала S характеризуется средним числом пакетов, передаваемых в одном окне. Очевидно, что $S \leq 1$.

Пропускная способность моноканала при СДПС оценивается следующим образом. Столкновения исключаются, если в течение периода $2T$, называемого периодом уязвимости, передается только один пакет (рис. 6,9). Если в течение периода уязвимости будет передаваться еще один пакет, который начинается либо в первом окне, либо во втором, происходит столкновение. Для наиболее простой оценки предположим, что поток запросов на передачу создается бесконечным числом систем, работающих независимо друг от друга и в результате этого порождающих пуассоновский поток запросов с суммарной интенсивностью G запросов на одно окно. Вероятность передачи пакета без столкновения определяется вероятностью поступления в период уязвимости только одного пакета и равна $q = e^{-2G}$. Следовательно, только q -я часть пакетов будет передана без искажений и интенсивность потока неискаженных пакетов.

$$S = Ge^{-2G} \quad (6.1)$$

Значение S характеризует пропускную способность моноканала, зависимость которой от интенсивности потока запросов G представлена на рис. 6.10 кривой *СДПС*. Максимум пропускной способности достигается при $G=0,5$ запросов на окно и составляет

$$S = 1/(2e) \approx 0,184 \text{ пакета на одно окно.} \quad (6.2)$$

Таким образом, СДПС позволяет использовать для передачи данных не более 18,4 % пропускной способности канала.

При конечном числе систем M пропускная способность моноканала

$$S = G(1 - G/M)^{M-1} \quad (6.3)$$

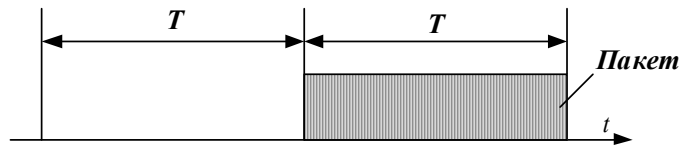


Рис. 6.9. Период уязвимости пакета при СДПС

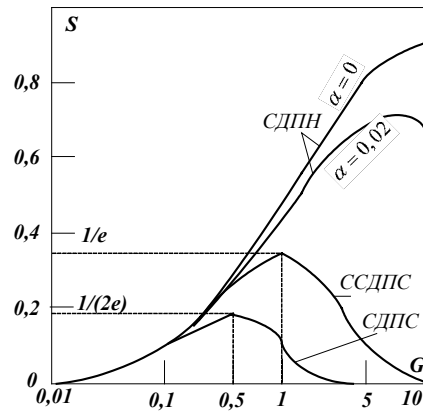


Рис. 6.10. Скорость передачи пакетов

При $M \rightarrow \infty$ эта зависимость принимает вид (6.1).

Задержка пакетов зависит от числа попыток передачи пакета, от задержки повторной передачи и является случайной величиной. Отношение $q = S/G$ характеризует долю пакетов, передаваемых без искажений, т. е. вероятность передачи без столкновений. Число попыток до успешной передачи – геометрически распределенная случайная величина $p_n = (1-q)^{n-1}q$, $n = 1, 2, \dots$. Величина p_n – вероятность передачи пакета с n -й попытки. Среднее число попыток

$$\bar{n} = 1/q = G/S \quad (6.4)$$

и средняя задержка пакета при передаче через моноканал

$$U = \bar{n}\bar{\tau} + T = \frac{G}{S}\bar{\tau} + T \quad (6.5)$$

где τ – средняя задержка повторной передачи.

Обычно значение $\bar{\tau}$ представляют в нормированном виде – в числе окон длительностью T . В таком случае нормированная средняя задержка пакета, определяемая числом окон,

$$U_H = \frac{G}{S}\bar{\tau}_H + 1 \quad (6.6)$$

где $\bar{\tau}_H$ – нормированная задержка повторной передачи.

Обратим внимание на зависимость интенсивности запросов G от задержки передачи: чем меньше $\bar{\tau}_H$, тем больше интенсивность запросов G , поскольку каждая система с уменьшением $\bar{\tau}_H$ начинает чаще обращаться к каналу. Если попытаться уменьшить задержку, интенсивность запросов возрастет и скорость передачи пакетов S будет отставать от скорости запросов на передачу данных. Это приведет к тяжелым последствиям: возрастет число пакетов, ожидающих передачи, увеличится задержка передачи пакетов, а скорость передачи будет уменьшаться до весьма малых значений, в результате чего система потеряет устойчивость. Чтобы избежать этого, необходимо ограничивать

интенсивность потока запросов (чтобы она не превышала 0,5 пакета на окно), за счет увеличения задержки на повторную передачу пакета. Более детально устойчивость систем со свободным доступом обсуждается ниже.

Таким образом, СДПС приводит к значительной потере пропускной способности канала. Однако этот способ прост в реализации, поскольку обнаружение столкновений и формирование случайной задержки на повторную передачу обеспечивается весьма простыми средствами. Простота реализации приводит к повышению надежности адаптеров, а, следовательно, и моноканала в целом. Существенный недостаток способа – возможность потери устойчивости из-за пульсаций потока запросов на передачу пакетов.

Устойчивость моноканала со свободным доступом. При свободном доступе к каналу зависимость числа пакетов, ожидающих передачи, от скорости передачи пакетов по каналу и средней задержки повторной передачи имеет вид, представленный на рис. 6.11. Скорость передачи пакетов ограничена предельным значением (для СДПС – значением 0,184 пакета на окно). При фиксированной задержке повторной передачи увеличение интенсивности поступления пакетов приводит к увеличению пропускной способности канала. При этом число пакетов m , ожидающих передачи, в том числе и повторной, оказывается незначительным и задержка доставки пакетов также невелика. Однако если в не который момент времени скорость поступления пакетов превысит предельную пропускную способность, канал переходит в режим, при котором число пакетов m , ожидающих передачи, и, следовательно, задержка передачи пакетов принимает большие значения. В этом режиме из-за большого числа пакетов, ожидающих пере дачи, увеличивается вероятность

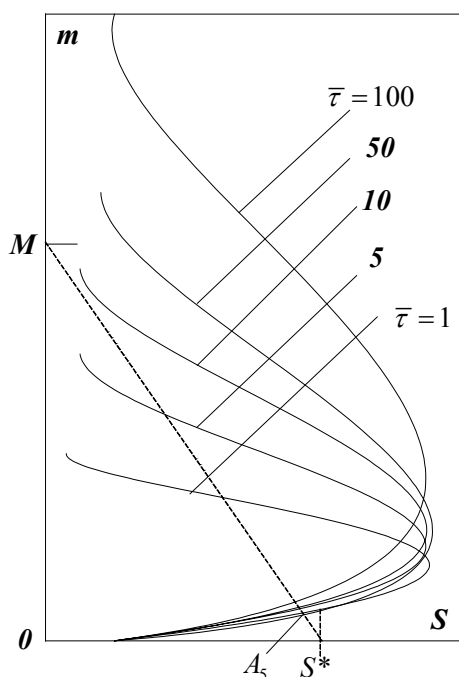


Рис. 6.11. Число пакетов в очереди на передачу

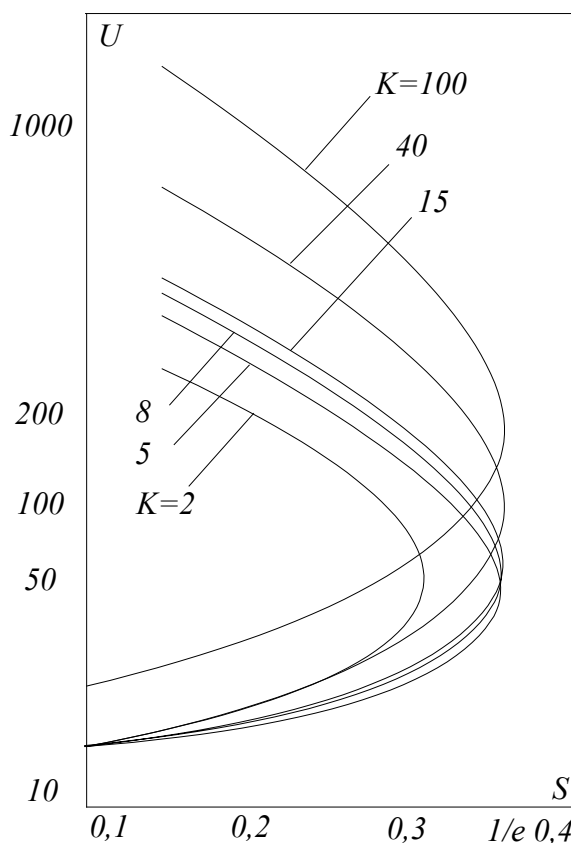


Рис. 6.12. Задержка передачи пакетов при СДПС

Основываясь на представленной зависимости $m = f(S, \bar{\tau}_H)$, оценим эффект от пульсации потока запросов на передачу пакетов. Пусть к каналу подключено M систем, из которых пакеты на передачу поступают с интенсивностью S^* . При этом каждая система порождает новый пакет с вероятностью $\sigma = S^*/M$. Нагрузку на канал в произвольный момент времени t будем характеризовать вектором $[m(t), s(t)]$, где $m(t)$ – число пакетов (систем), ожидающих повторной передачи, и $s(t)$ – скорость поступления новых пакетов, которую будем характеризовать числом пакетов, поступающих для передачи в течение окна. Нагрузка на канал $[m(t), s(t)] = [m, (M - m)\sigma]$, где $(M - m)\sigma$ – число новых пакетов, порождаемых $(M - m)$ системами, каждая из которых в течение окна может получить пакет с вероятностью a . С увеличением числа пакетов m , ожидающих передачи, скорость поступления новых пакетов $s(t)$ линейно уменьшается, поскольку число систем конечно. Линия $m = M - S/\sigma$, изображенная на рис. 6.11 штрихами, называется линией нагрузки канала. Если в момент времени t скорость поступления новых пакетов $s(t) = S^*$, то нагрузка канала $[m(t), s(t)] = [0, S^*/\sigma] = [0, M]$. Если же скорость поступления пакетов упала до нуля, то нагрузка $[m(t), s(t)] = [M, 0]$. Пульсация потока пакетов, поступающих в моноканал для передачи, может интерпретироваться перемещением точки (m, S) по линии нагрузки. Точка пересечения линии нагрузки с кривой $m = f(S, \bar{\tau}_H)$ называется рабочей

$$m = f(S, \bar{\tau}_H)$$

хотя бы в одной точке, и неустойчив, если они не соприкасаются. Устойчивый режим характеризуется тем, что увеличение числа пакетов, ожидающих передачи, сопровождается увеличением пропускной способности канала, за счет чего очередь будет «рассасываться». При неустойчивом режиме увеличение числа пакетов в очереди на передачу сопровождается снижением пропускной способности канала, в результате чего канал переходит в нерабочее состояние, при котором все M систем хранят пакеты для передачи и из-за столкновений ни один пакет не может быть передан. Из этого состояния сеть может быть выведена только путем внешнего вмешательства.

Из рис. 6.11 видно, что устойчивость канала обеспечивается за счет согласования интенсивности потока пакетов S^* , генерируемого системами, с предельной пропускной способностью канала, а также за счет выбора времени задержки повторной передачи. Интенсивность потока пакетов S^* должна быть меньше предельной пропускной способности канала, и чем она меньше, тем меньше вероятность потери устойчивости при пульсации нагрузки. Среднюю задержку на повторную передачу следует увеличивать, в результате чего снизится интенсивность повторных передач пакетов при столкновениях.

Синхронный свободный доступ с проверкой столкновений (ССДПС). Способ ССДПС применяется в основном в магистральных структурах. Для уменьшения периода уязвимости работа систем синхронизируется – все системы начинают передачу в один и тот же момент времени. Период синхронизации равен длительности окна T , используемого для передачи одного пакета. В результате синхронизации период уязвимости уменьшается до одного окна. Если две и более системы начинают передавать пакеты и в одном окне, происходит столкновение пакетов, свидетельствующее о невозможности передачи пакетов в текущем окне. После этого каждая система случайным образом вырабатывает задержку $\tau_H = 1, \dots, T_{\max}$. Система повторно передает пакет в окне, следующем через $(\tau_H - 1)$ окон за текущим. Если вновь происходит столкновение пакетов, передача задерживается на τ_H окон и процесс продолжается до выполнения передачи без столкновения.

Для ССДПС скорость передачи пакетов по Моноканалу при бесконечном числе систем

$$S = Ge^{-G} \quad (6.7)$$

а при числе систем M

$$S = G(1 - G/M)^{M-1} \quad (6.8)$$

Зависимость (6.7) представлена на рис. 6.10 кривой ССДПС. Максимальная скорость передачи пакетов достигается при $G = 1$ запрос на окно и составляет

$$S = 1/e \approx 0,368 \text{ пакета на окно.} \quad (6.9)$$

Таким образом, при ССДПС пропускная способность моноканала увеличивается в 2 раза по сравнению с СДПС, но составляет менее 37 % пропускной способности физического канала. Средняя задержка передачи пакета определяется по формулам (6.4), (6.5), причем нормированная средняя задержка повторной передачи $\bar{\tau}_{II} = \tau_{\max} / 2$. Зависимость задержки от скорости передачи пакетов по моноканалу представлена на рис. 6.12.

При ССДПС существенно повышается по сравнению с СДПС пропускная способность моноканала, но требуются генератор синхронизирующих сигналов и линии для их передачи, в результате чего увеличиваются затраты, оборудования и снижается надежность моноканала.

Свободный доступ с проверкой несущей (СДПН). Система, имеющая пакет для передачи, перед тем как начать передачу, проверяет состояние моноканала – наличие в нем сигналов, используемых для передачи данных (сигналы несущей частоты, потенциальные или импульсные). Операция проверки состояния канала называется *проверкой несущей*. Если канал свободен, адаптер системы начинает передачу пакета. Если канал занят, передача откладывается, на время со средним значением, например, $T/2$, где T – время передачи пакета по моноканалу. За счет проверки несущей вероятность столкновений существенно уменьшается (поскольку происходят они только в том случае, если две системы начинают передачу практически одновременно). В результате этого увеличивается степень использования пропускной способности канала, т. е. скорость передачи данных.

Скорость передачи данных при СДПН оценивается следующим образом. Предположим, что для всех пар систем сети время распространения сигнала одинаково и равно a . Это время влияет на вероятность столкновения пакетов. Если одна система начинает передачу в момент времени t , а другая система в момент времени $t + \delta$, где $\delta < a$, происходит столкновение пакетов, поскольку в течение времени a после начала передачи занятый канал воспринимается: любой системой как свободный. Вероятность того, что ни одна система не начнет передачу в течение интервала a , равна $e^{-\alpha G}$, где $\alpha = a/T$ – нормированное время распространения сигнала по моноканалу. Скорость передачи по моноканалу

$$S = \frac{Ge^{-\alpha G}}{G(1 + 2\alpha) + e^{-\alpha G}} \quad (6.10)$$

При $\alpha \rightarrow 0$

$$\lim_{\alpha \rightarrow 0} S = \frac{G}{G+1} \quad (6.11)$$

Зависимость (6.10) для $\alpha = 0$ и $\alpha = 0,02$ представлена на рис. 6.10. При малом времени распространения сигнала по каналу ($\alpha = T$) с увеличением интенсивности потока запросов на передачу пакетов G скорость передачи пакетов по моноканалу возрастает, приближаясь к пропускной способности физического канала. Нормированная средняя задержка при передаче пакета

$$U_H = (1 + \alpha) + (G/S - 1)(2\alpha + 1 + \bar{\tau}_H) \quad (6.12)$$

Проверка несущей реализуется достаточно простой схемой, встраиваемой в сетевой адаптер, и приводит к существенному повышению пропускной способности моноканала. Однако СДПК не исключает возможности столкновения пакетов из-за конечности времени распространения сигналов a , в течение которого две или более системы могут начать передачу пакетов. Последствия столкновения пакетов ликвидируются двумя способами. Во-первых, в адаптеры систем можно встраивать схемы проверки столкновений. В этом случае доступ к каналу осуществляется с проверкой несущей и столкновений (СДПНС). Во-вторых, для ликвидации последствий столкновений можно использовать механизм квитанций и перезпроса. При этом передающая система не контролирует столкновения и возможно искажение пакетов. Принимающая система проверяет поступивший пакет с помощью контрольного суммирования к передаваемой с пакетом проверочной последовательности. Если ошибки в пакете не обнаружены, передающей системе направляется квитанция о приеме пакета. При отсутствии квитанции по истечении тайм-аута передающая система вновь направляет пакет адресату.

Эстафетный доступ. В магистральных структурах эстафетный доступ реализуется по схеме, приведенной на рис. 6.13. Сетевые адаптеры систем, подключенные к моноканалу, связаны кольцевой цепью, по которой между адаптерами передается эстафета – сигнал, разрешающий доступ к моноканалу. Если в адаптере пакет на передачу отсутствует, этот адаптер передает эстафету следующему адаптеру. Если адаптер хранит пакет для передачи, то по прибытии эстафеты адаптер начинает передачу пакета в канал и по окончании передачи пересылает эстафету следующему адаптеру. При эстафетном доступе почти полностью используется пропускная способность канала. Время доставки пакета не превышает NT , где N – число активных систем в сети и T – время передачи пакета (кадра) по каналу. Все системы находятся в одинаковых условиях и получают право на передачу с частотой не ниже $1/(NT)$.

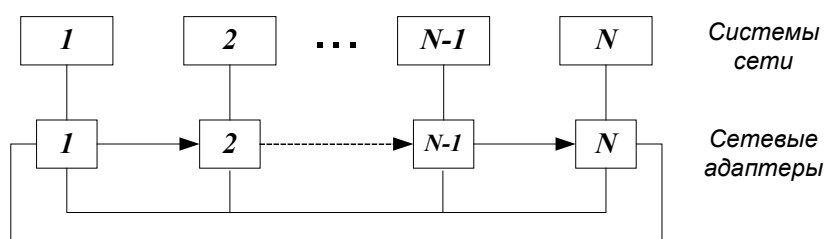


Рис. 6.13. Эстафетный доступ к каналу в ЛВС с магистральной структурой

ПНК	Адрес получателя	Адрес отправителя	Данные	ПКК
-----	------------------	-------------------	--------	-----

Рис. 6.14. Структура кадра

В кольцевых структурах эстафетный доступ к каналу реализуется с использованием эстафеты – маркёра, последовательно передаваемого по кольцу от одной системы к другой. В качестве маркёра используется специально выделенный для этого разряд пакета – бит в

последовательности начала кадра *ПНК* (рис. 8.14). Если система находится в состоянии готовности к передаче пакета к приходит маркёр, она изымает его из кольца и направляет в канал кадр, пример структуры которого приведен на рис. 6.14. Кадр поступает в следующую систему сети, которая ретранслирует его к очередной системе, и т. д. Каждая система сравнивает свой собственный адрес с адресом получателя, указанным в кадре. Если адреса совпадают, система принимает пакет в свою память и одновременно транслирует его дальше. Факт приема пакета системой отмечается установкой в 1 специального бита приема, выделяемого для этой цели в последовательности конца кадра *ПКК*. Передающая система находится в состоянии передачи до возвращения в нее отправленного кадра. Принимая ранее переданный кадр, передающая система сравнивает его с хранимым в памяти текстом, анализирует значение бита приема и, установив правильность передачи и факт приема кадра, посылает маркёр следующей системе, которая по получении маркёра имеет право на передачу своего кадра. Если в системе отсутствуют данные для передачи, она передает маркёр следующей системе сети. Таким образом, маркёр последовательно передается между системами кольца, поочередно предоставляя им право на передачу данных.

При начальном запуске ЛВС необходимо сформировать маркёр. Функция формирования маркёра может быть возложена на одну главную систему или на несколько систем сети. В результате помех, воздействующих на канал, и отказов систем маркёр может быть потерян. Факт потери устанавливается главной системой с помощью тайм-аута, длительность которого равна максимальной продолжительности передачи маркёра по кольцу. Если маркёр потерян, главная система генерирует его повторно в автоматическом режиме или под управлением оператора.

Управление информационным каналом. Информационный канал ЛВС строится на основе физического канала, дополняя последний средствами обеспечения достоверности данных, передаваемых в форме пакетов между узлами ЛВС. Порядок функционирования информационного канала задается протоколом управления, который относится ко второму подуровню уровня 2 (см. рис. 6.2) и определяет формат пакетов, средства контроля данных и исправления ошибок, вносимых в данные при передаче пакета по каналу. В ЛВС используется дейтаграммный способ передачи данных, т. е. каждый пакет рассматривается как независимый объект, передаваемый между узлами сети. Пропускная способность моноканала, как правило, превышает потребную для сети, и поэтому нет необходимости в уменьшении размеров служебных полей. В связи с этим в ЛВС обычно используется единственный формат пакета с фиксированным размещением полей, что упрощает процедуры и средства формирования и приема пакетов (кадров). Структура кадра представлена на рис. 6.14. Последовательность начала кадра состоит, как правило, из 2–16 битов, используемых в качестве флага начала пакета и поля управления доступом к моноканалу. Поле данных в различных ЛВС имеет разную длину – обычно от 4 до 128, а в отдельных сетях до 512 байт. Корректность данных, содержащихся в пакете, может контролироваться разными способами: по четности байтов или с использованием 16- и 32-разрядных циклических сумм. В последовательности конца кадра выделяются поля для кода циклической суммы, признака доставки пакета, признака продолжения сообщения в следующем пакете и др.

Система, принимающая пакет, проверяет его корректность с помощью средств контроля достоверности данных. При обнаружении ошибки в пакете данных производится повторная передача пакета в порядке, определяемом протоколом управления информационным каналом, например с использованием квитанции и тайм-аута. В кольцевых сетях пакет возвращается отправителю с отметкой о приеме, т. е. несет в себе квитанцию об успешном приеме. Если отметка о приеме отсутствует, система повторно передает пакет и выполняет это действие заданное число раз, определенное протоколом. Если попытки доставить пакет оказываются безуспешными, фиксируется неработоспособность (недоступность) адресата и взаимодействие прекращается. В магистральных сетях прием

подтверждается квитанцией. При отсутствии квитанции производится повторная передача пакета по истечении тайм-аута.

6.3. АДАПТЕРЫ

Сетевые адаптеры обеспечивают сопряжение узлов ЛВС с моноканалом и реализуют протоколы канального уровня: управления физическим каналом, доступом к каналу и информационным каналом. Структура адаптера представлена на рис. 6.15. Приемопередатчик согласует логические сигналы, формируемые в адаптере, с физическими сигналами в моноканале – уровнями сигналов в витых парах, биполярными сигналами в коаксиальном кабеле и световыми сигналами в волоконно-оптической линии и тем самым реализует управление физическим каналом. Блок управления доступом выполняет протокол, доступа к моноканалу, взаимодействуя с ним через приемопередатчик. Блок управления передачей обеспечивает вывод на приемопередатчик последовательности битой, соответствующих пакету. Блок управления приемом анализирует пакеты, передаваемые через моноканал, и выделяет пакеты, адресованные узлу, обслуживаемому адаптером. Блоки управления передачей и приемом либо имеют собственную буферную память для хранения пакетов, либо используют память ЭВМ. Четыре указанных блока образуют сетевую часть адаптера которая связывается с ЭВМ с помощью блока сопряжения через соответствующий интерфейс ввода – вывода.

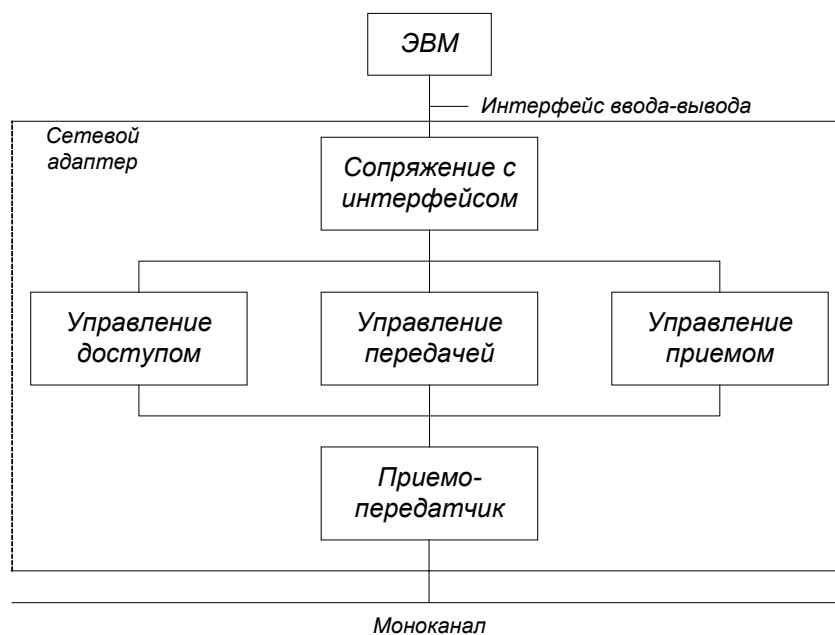


Рис. 6.15. Сетевой адаптер

Приемопередатчики. Способ построения приемопередатчиков оказывает значительное влияние на характеристики сети, и в первую очередь на надежность. При использовании в моноканале витой пары и коаксиального кабеля надежность канала (вероятность искажения передаваемых сигналов и устойчивость к отказам узлов сети) существенно зависит от организации электропитания и заземления приемопередатчиков. Радикальный способ развязки элементов сети по питанию – установка в соответствующих цепях развязывающих элементов – трансформаторов или оптронов (пар из светодиода и фотодиода), разрывающих электрические связи между источниками и приемниками сигналов. В магистральных ЛВС развязывающие элементы устанавливаются между приемопередатчиком и остальной частью адаптера (рис. 6.16, а), а в кольцевых сетях – в конце каждого сегмента кабеля, соединяющего соседние адаптеры (рис. 6.16, б).

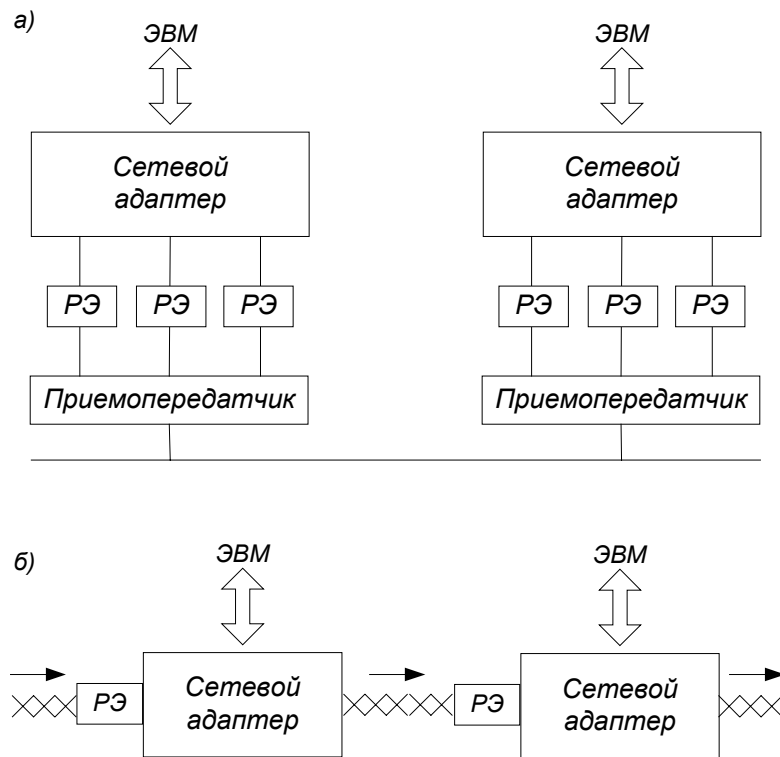


Рис. 6.16. Развязка элементов ЛВС по питанию и заземлению

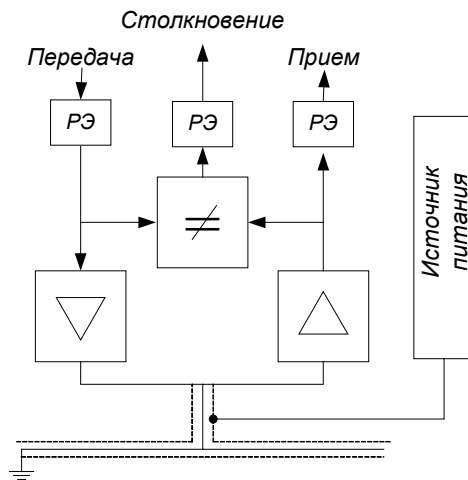


Рис. 6.17. Приемопередатчик ЛВС с магистральной структурой

Пример приемопередатчика для сети с магистральной структурой приведен на рис. 6.17. В качестве передающей среды используется коаксиальный кабель, к которому подключаются усилители передатчика и приемника. Для выявления столкновений пакетов используется схема сравнения, на выходе которой при несовпадении сигналов, поступающих на передатчик и с приемника, формируется сигнал о столкновении. Для электрической развязки приемопередатчика и остальной аппаратуры адаптера используются оптрона. В данной схеме приемопередатчик имеет собственный источник питания, изолированный от земли. Нулевая шина (земля) приемопередатчика соединяется с экраном кабеля, что существенно снижает помехи, возникающие на протяженной линии. Приемопередатчики магистральных сетей должны иметь высокое сопротивление по отношению к каналу, чтобы

изменение числа систем (адаптеров), подключаемых к каналу, не вызывало значительного изменения нагрузки.

Пример приемопередатчика для сети с кольцевой структурой приведен на рис. 6.18. Сегмент канала, выполненный из экранированной скрученной пары, соединяет передатчик одного адаптера с приемником следующего адаптера через развязывающий элемент. На выходе приемника формируются логические сигналы, представляющие принимаемые биты. Кроме того, сигналы, поступающие на приемник, используются для выработки сигналов синхронизации, обеспечивающих тактирование схем адаптера. Поступившие в адаптер сигналы должны ретранслироваться в следующий адаптер сети. Для обработки поступающей информации (установки бита приема в единицу) в цепь ретрансляции введена схема задержки битов. Обычно задержка соответствует 2–4 бит. Селектор служит для управления работой передатчика, через который на разных этапах работы адаптера могут ретранслироваться поступающие на вход адаптера данные или выводиться биты передаваемого кадра. В кольцевой сети каждый адаптер, т.е. его приемопередатчик, должен обеспечивать трансляцию сигналов даже при выключении ЭВМ, обслуживаемой адаптером. Поэтому в сети имеется специальный источник питания, к которому подключаются только приемопередатчики сетевых адаптеров.

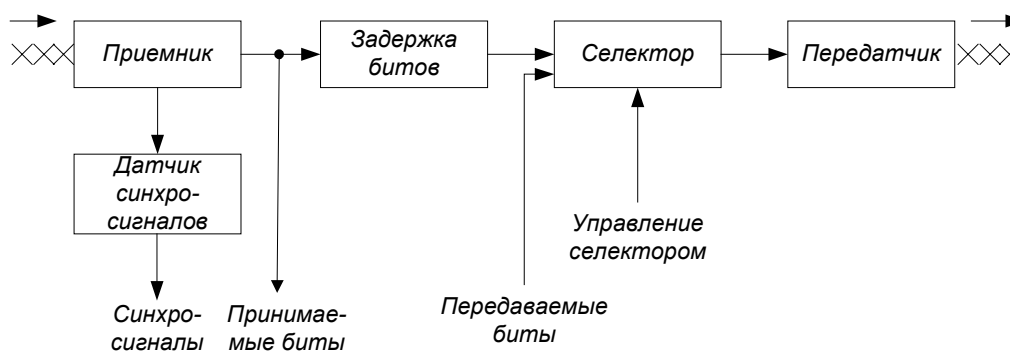


Рис. 6.18. Приемопередатчик ЛВС с кольцевой структурой

Отказ любого приемопередатчика в кольцевой сети нарушает работоспособность всей сети. Для обеспечения надежности сети принимаются специальные меры: устанавливаются обходные реле, замыкающие вход приемника на выход передатчика при отключении и отказе ЭВМ и адаптера, или кольцо дублируется, т. е. каждый адаптер снабжается двумя приемопередатчиками и для соединения систем используются два кабеля (моноканала).

Управление доступом и информационным каналом. Как правило, в сетевом адаптере реализуются функции, обеспечивающие доступ к каналу, прием и передачу пакета, вычисление и проверку контрольных сумм, а функции, связанные с управлением информационным каналом, возлагаются на программные средства ЭВМ, обслуживаемой адаптером. Блок сопряжения с интерфейсом ЭВМ обеспечивает передачу данных и сигналов прерывания между ЭВМ и адаптером.

Наиболее существенное влияние на организацию адаптеров оказывает способ обмена данными между адаптерами и ЭВМ. Могут использоваться два способа: без буферизации и с буферизацией пакетов. По первому способу пакет, подлежащий передаче, размещается в оперативной памяти ЭВМ, откуда он пересылается в адаптер в виде последовательности слез. Адаптер преобразует каждое поступившее слово в последовательность битов. После передачи битов слова в моноканал адаптер через интерфейс ЭВМ обращается к оперативной памяти ЭВМ за следующим словом. При приеме пакета в адаптере из поступающих битов формируются слова, которые последовательно отсылаются в заданную область оперативной памяти ЭВМ в темпе формирования слов. По второму способу в адаптере организуются буферные запоминающие устройства для хранения передаваемых и принимаемых пакетов. Передаваемый пакет сначала выводится из оперативной памяти ЭВМ и в буфер адаптера,

откуда он передается в моноканал. При приеме в буфере накапливается пакет, который по окончании приема передается через интерфейс ЭВМ в заданную область оперативной памяти ЭВМ.

Выбор способа обмена данными между адаптером и ЭВМ зависит от соотношения пропускной способности интерфейса ЭВМ и моноканала. Если пропускная способность интерфейса меньше, адаптер должен иметь буферную память. В противном случае необходимость в буферизации пакетов отсутствует.

Определим в общих чертах структуру управления передачей и приемом кадров (рис. 6.19). При этом предполагается, что кадр имеет структуру, изображенную на рис. 6.14.

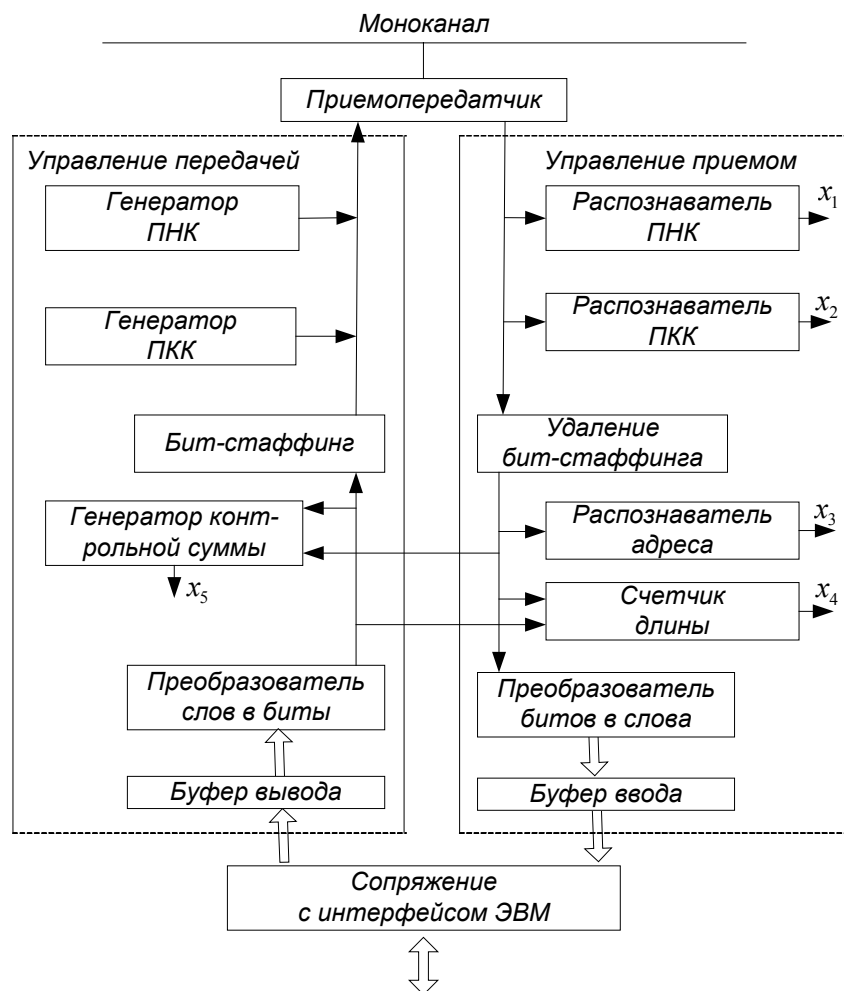


Рис. 6.19. Управление приемом и передачей в адаптере ЛВС

Управление передачей кадров реализуется следующим образом. На ЭВМ в блок сопряжения с интерфейсом перед началом передачи кадра заносится адрес начала области оперативной памяти (в которой хранится пакет), а также длина пакета в битах. Блок сопряжения считывает из оперативной памяти первое слово пакета, которое передается в буфер вывода и затем в преобразователь параллельного кода в последовательный, содержащий регистр для хранения одного слова. После освобождения буфера вывода в него передается из оперативной памяти очередное слово. Передача кадра начинается с посылки в передатчик последовательности начала кадра, формируемой специальным генератором. Вслед за ней через схему бит-стаффинга, обеспечивающую прозрачность физического канала, выводится последовательность битов, составляющих пакет. При этом преобразователь параллельного кода в последовательный по окончании передачи хранимого в нем слова загружается очередным словом из буфера вывода, а в последний вводится новое слово пакета, считываемое из оперативной памяти. Процесс продолжается до тех пор, пока

не будет передано заданное число битов, что отмечается сигналом x_4 , формируемым счетчиком длины. При передаче данных определяется контрольная сумма, которая передается вслед за данными, после чего на передатчиках выводится последовательность конца кадра, формируемая генератором ПКК.

Готовность адаптера к приему кадра обеспечивается передачей адреса оперативной памяти, выделенной для размещения принимаемого пакета из ЭВМ в блок сопряжения. Сигналы с приемника обрабатываются блоком управления следующим образом. Распознаватель последовательности начала кадра формирует сигнал x_1 , отмечающий начало пакета, передаваемого в кадре. Последующие биты, поступающие из приемника, обрабатываются схемой удаления бит-стаффинга и поступают на распознаватель адреса, сравнивающий адрес получателя с собственным адресом адаптера. Совпадение адресов отмечается сигналом x_3 , определяющим принадлежность передаваемого кадра данному адаптеру. Принимаемые биты поступают на преобразователь последовательного кода в параллельный. Сформированное преобразователем слово передается в буфер вывода, из которого оно записывается в соответствующую ячейку оперативной памяти. Биты, составляющие пакет, формируются на счетчике длины и обрабатываются генератором контрольной суммы. Счетчик фиксирует длину принятого пакета и формирует сигнал x_4 , если длина пакета превосходит предельную допустимую. При обнаружении ошибки в пакете генератор контрольной суммы вырабатывает сигнал x_5 . Прием кадра и передача содержащегося в нем пакета в оперативную память заканчивается при поступлении последовательности конца кадра, что отмечается сигналом x_2 . Сигнал x_4 , формируемый счетчиком длины, используется для прекращения приема пакета, имеющего недопустимую длину.

Блок сопряжения с интерфейсом ЭВМ по окончании передачи и приема кадра передает в ЭВМ слово состояния адаптера, содержащее необходимую информацию о ходе передачи и приема кадра.

6.4. РАСШИРЕНИЕ И КОМПЛЕКСИРОВАНИЕ

Расширение ЛВС направлено на увеличение протяженности сети и числа узлов, объединяемых сетью, что достигается увеличением длины моноканала. Комплексование сетей – информационное объединение нескольких сетей через средства передачи данных.

Способы расширения. В ЛВС с магистральной структурой вводится ограничение на предельную длину кабеля и число подключаемых к магистрали систем (приемопередатчиков), за счет чего достигается требуемая пропускная способность магистрали при умеренных затратах оборудования в приемопередатчике. Для увеличения протяженности магистрали используются *ретрансляторы (повторители)* (рис. (6.20, а), обеспечивающие восстановление электрических и временных параметров сигналов. При этом нагрузка на любой передатчик ограничивается сегментом кабеля и подключенными к нему приемниками, в том числе приемником ретранслятора. Приемники остальных сегментов кабеля являются нагрузкой только для ретранслятора. Возможности расширения сети с помощью ретрансляторов ограничиваются пропускной способностью канала и временем распространения сигнала между наиболее удаленными станциями ЛВС, предельное допустимое значение которого определяется протоколам» управления физическим каналом и доступом к каналу. Предельные размеры сети ограничены максимальным допустимым числом ретрансляторов и расстоянием между наиболее удаленными станциями.

Для связи сегментов канала могут использоваться ретрансляторы длинной линии (рис. 6.20,б), с помощью которых возможна передача сигналов на расстояния 10^2 – 10^3 м.

В ЛВС с кольцевой структурой электрические ограничения на число узлов, объединяемых в кольцо, отсутствуют и ограничивается лишь предельная длина сегмента, соединяющего соседние узлы. Вместе с тем при увеличении числа узлов в кольцевой сети возрастает задержка передачи пакетов и увеличивается время доставки пакетов. Для исключения этого эффекта расширение кольцевых ЛВС производится на основе многоканальной структуры сетей.

Способы комплексирования. Для расширения кольцевых ЛВС используются двух- и трехуровневые кольцевые структуры (рис. 6.21). В этом случае сеть строится из нескольких кольцевых подсетей, а каждой из которых передача данных происходит в обычном порядке. Для объединения подсетей используется дополнительное кольцо, содержащее коммутационные узлы КУ, через которые кольца нижнего уровня подсоединяются к кольцевому каналу более высокого уровня. Коммутационные узлы принимают пакеты, адресованные узлам других колец, и направляют их в соответствующие коммутационные узлы для передачи адресату. Коммутация пакетов производится по таблицам маршрутизации, хранимым в памяти коммутационных узлов. За счет иерархических кольцевых структур можно существенно увеличить число станций, входящих в сеть, причем средняя задержка незначительно увеличивается. К тому же при преобразовании кольцевой структуры в оптимальную двух- или трехкольцевую повышается надежность сети.

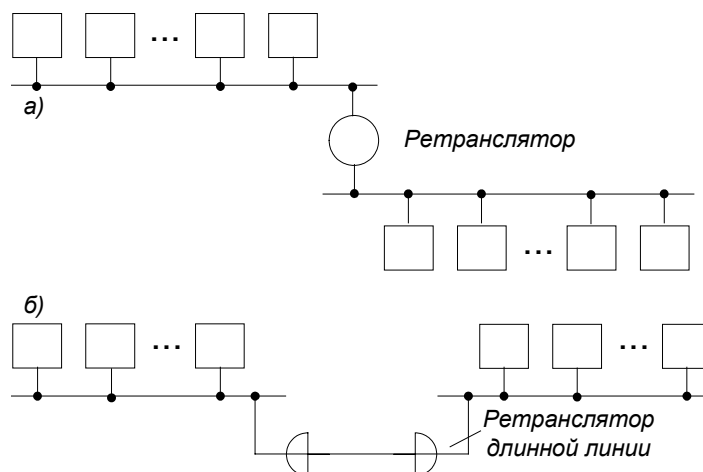


Рис. 6.20. Расширение магистральной ЛВС с помощью ретрансляторов

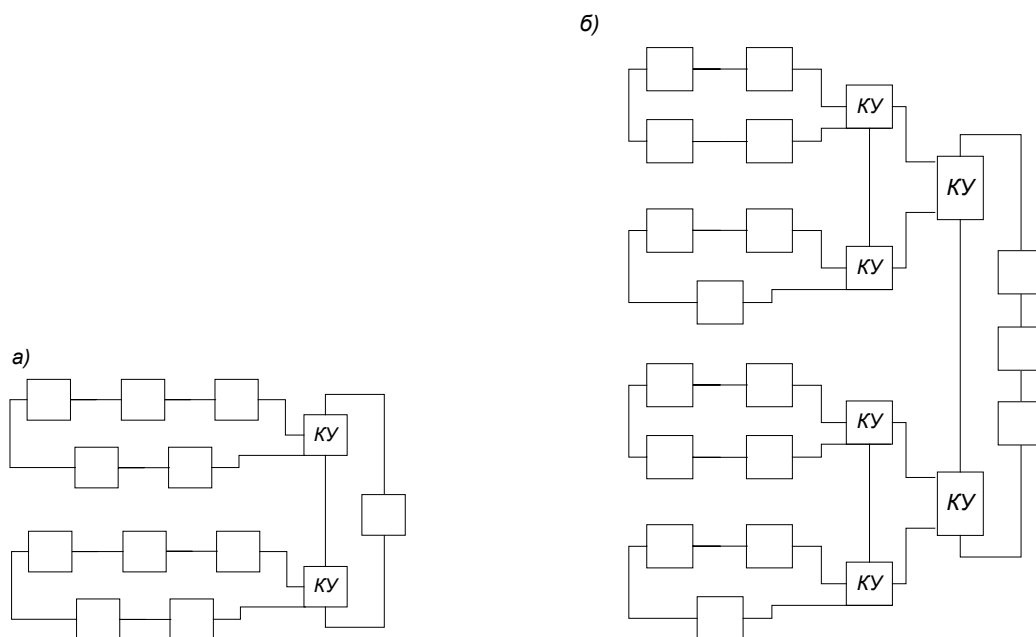


Рис. 6.21. Двух- и трехуровневая кольцевая структура ЛВС

Комплексирование моноканальных сетей в сеть более высокого уровня может производиться с помощью межканальных станций *МКС* (рис. 6.22), соединяющих два канала. Межканальная станция выполняет функции, аналогичные функциям узла связи в сетях передачи данных. Она селектирует передаваемые по моноканалу пакеты, выделяя из них те, которые адресованы абонентам, подключенным к выходному каналу станции, принимает выделенные пакеты, хранит их в буферной памяти и передает в выходной канал, связывающий *МКС* с адресатами. При этом необходимость в

коммутации пакетов отсутствует, поскольку *МКС* имеет только одно выходное направление. Межканальная станция, связывающая два моноканала одного типа, называется мостом. Если сопрягаемые сети находятся на значительном расстоянии, для связи между ними можно использовать линию передачи данных. В этом случае сопряжение сетей производится с помощью двух *МКС*, связанных линией передачи данных, – длинного моста. Межканальная станция длинного моста преобразует протокол моноканала в протокол передачи данных, например, HDLC, и обратно.

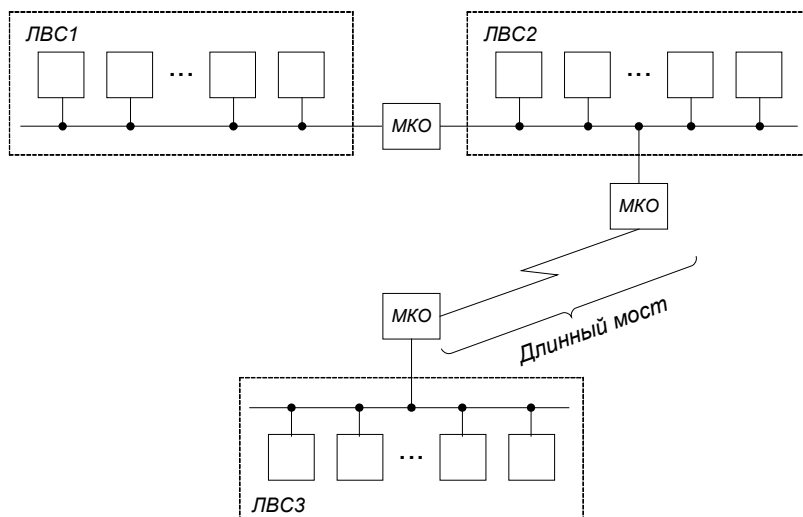


Рис. 6.22 Многоканальная ЛВС

Многоканальная сеть может быть однородной, включающей в себя моноканалы одного типа, а также неоднородной и объединять сети с магистральной и кольцевой структурой и разными протоколами управления физическим каналом, доступом к каналу и информационным каналом. В последнем случае, кроме селекции и управления каналами, необходимо преобразование протоколов информационных каналов. Для этих целей используется интерфейсная система, иначе называемая шлюзом. Интерфейсная система строится на базе микро-ЭВМ, реализующей функции преобразования протоколов. К микро-ЭВМ подключаются адаптеры для сопряжения ЭВМ с моноканалами. С помощью интерфейсной системы возможно сопряжение ЛВС и СПД с использованием протокола X25, т. е. объединение локальных и глобальных вычислительных сетей.

6.5. РЕАЛИЗАЦИЯ

Разработка ЛВС началась в 70-х годах. К настоящему времени создано большое число вариантов ЛВС, отличающихся типом передающей среды, пропускной способностью моноканала, структурой и назначением. Типичные области применения ЛВС – системы административного управления (учрежденческие), автоматизации проектирования и технологической подготовки производства, автоматизации научных исследований, промышленные системы (системы управления технологическими процессами и производством) и бортовые. К последним предъявляются требования повышенной надежности при работе в условиях сильных электромагнитных помех и, возможно, в широком диапазоне температур окружающей среды.

Ниже описываются три реализации ЛВС, наиболее ярко представляющие основные аспекты организации сетей.

Сеть Ethernet. Эта сеть разработана в середине 70-х годов фирмой «Ксерокс» (США), и ее архитектура используется в качестве типовой фирмами DEC и «Интел», производящими мини- и микро-ЭВМ. Основное назначение сети – системы административного управления.

Сеть имеет магистральную структуру. В качестве передающей среды используется коаксиальный кабель, передача данных по которому выполняется со скоростью 10 Мбит/с. Сеть может объединять до 1024 систем при максимальном расстоянии между ними 2,5 км.

Структура сети представлена на рис. 6.23. В простейшем варианте (рис. 6.23, а) сеть состоит из сегмента (длиной до 500 м) коаксиального кабеля, к которому может подключаться до 100 приемопередатчиков, обслуживающих системы. Кабель приемопередатчика состоит из четырех витых пар для передачи, приема данных, передачи сигналов столкновения и питания. Максимальная длина кабеля приемопередатчика 50 м. В варианте на рис. 6.23, б сегменты объединяются в магистраль с помощью ретрансляторов; подключаемых к приемопередатчикам. Между любыми двумя системами не может быть более двух ретрансляторов. Пример сложной конфигурации представлен на рис. 6.23, в. Для связи удаленных сегментов используется ретранслятор длинной линии, предельная длина которой может достигать 1 км. При любой конфигурации расстояние между системами не должно превышать 2,5 км.

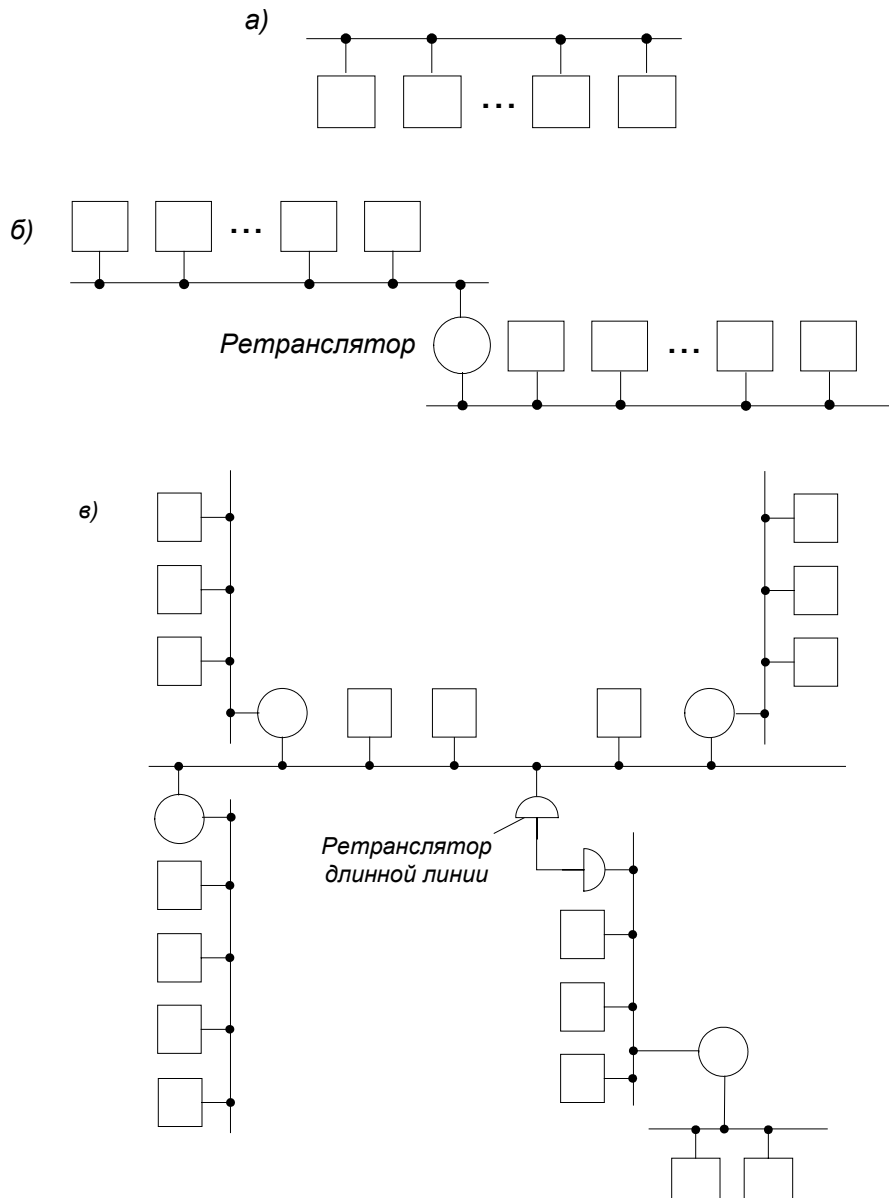


Рис. 6.23. Варианты ЛВС



Рис. 6.24. Формат кадра ЛВС Ethernet

Сеть имеет трехуровневую организацию. Верхний уровень – прикладной – объединяет функции и средства обработки данных и ввода–вывода сообщений. Канальный и физический уровни обеспечивают доступ к моноканалу и передачу по нему сообщений. Функции двух нижних уровней реализуются приемопередатчиком и сетевым адаптером, который в этой сети называется абонентской станцией. Приемопередатчик установлен непосредственно на коаксиальном кабеле и служит для формирования сигналов, приема сигналов из канала и выявления столкновений. С помощью кабеля к приемопередатчику подсоединяется плата контроллера – абонентской станции, которая реализует доступ к моноканалу, передачу и прием сообщений, а также сопряжение с интерфейсом ЭВМ – шиной ввода–вывода. ЭВМ выполняет прикладные функции, функции управления, передачей данных между взаимодействующими процессами (абонентами), а также часть функций по управлению информационным каналом.

Передаваемые по каналу данные представляются в манчестерском коде. Для исключения потерь первых битов кадра передача предваряется последовательностью из 64 бит:

1010101010101010 . . . 10101011

Последовательность 1010...10 – это серия сигналов с частотой 5 МГц. Код 11, завершающий последовательность, отмечает момент начала кадра. Приемник включается в работу при обнаружении последовательности 1010...10 и после поступления кода 11 начинает прием битов кадра. После исчезновения сигналов в канале кадр считается принятым и начинается его обработка.

Способ доступа к каналу – свободный с проверкой несущей и столкновений. Контроллер проверяет наличие сигналов в канале и после исчезновения несущей начинает передачу кадра в канал с задержкой 9,8 мкс. Задержка создает межкадровый интервал, в течение которого аппаратура контроллеров восстанавливает свое исходное состояние и может принимать очередной кадр. В том случае, если при передаче приемопередатчик выявляет столкновение, контроллер прекращает передачу кадра и «подкрепляет» столкновение, передавая в канал помеху в виде последовательности из 32–48 бит. При этом системе разрешается сделать еще 15 попыток передачи кадра. Если все 16 попыток оказались неудачными, выдается сигнал об ошибке.

Формат кадра сети «Ethernet» представлен на рис. 6.24. Адреса разделяются на три типа; физические, групповые и широковещательные. Физический – адрес, однозначно определяющий систему сети. Групповой адрес определяет группу систем. Широковещательный адрес относится ко всем системам сети. Тип адреса кодируется в первом бите адреса приемника: 0 – физический, 1 – групповой или широковещательный адрес. Последний кодируется единицами во всех 48 разрядах поля адреса приемника. Источник идентифицируется своим физическим адресом, представляемым во втором поле

кадра. Поле типа кадра идентифицирует команды и ответы. Контроль правильности передаваемых данных осуществляется с помощью 32-разрядного циклического кода.

Таблица 6.2. Структура пакета сети

Номер бита	Назначение	Номер бита	Назначение
1	Метка начала пакета	12-19	Адрес источника
2	Эстафета (метка «свободен-занят»)	20-35	Данные
3	Метка монитора	36-37	Метка приемника
4-11	Адрес приемника	38	Бит четности

Сеть Cambridge Ring (Кембриджское кольцо). Сеть разработана в Кембриджском университете в 1979 г. Аппаратура выпускается под названием Polynet. Сеть имеет кольцевую структуру. Сетевые адаптеры соединяются кабелем, состоящим из двух пар скрученных экранированных проводов. Обе пары используются для передачи сигналов. Значение 1 передается сигналами сразу по двум парам, значение 0 – сигналом по одной паре. Эти же игры используются для питания; цепи разных пар имеют различные потенциалы. Расстояние между соседними адаптерами не должно превышать 100 м, чтобы скорость передачи данных по моноканалу была не менее 10 Мбит/с при достаточно простых приемеопередатчиках.

Данные передаются по кольцу в виде пакетов из 38 бит. Структура пакета определена в табл. 6.2. Пакет может быть свободным или затолченным. Свободный пакет отмечается единичным значением во втором бите 2. При прохождении через адаптер, хранящий данные для передачи, свободный пакет заполняется данными и передается как занятый по кольцу от источника к приемнику. Адаптер, которому адресован пакет, отмечает (метка приемника) «пакет принят», «пакет отвергнут», «приемник занят». Пакет возвращается источнику, где сравнивается с переданным оригиналом, и при условии приема адресатом очищается от данных и возвращается в кольцо как свободный. Реагируя на метки «пакет отвергнут» и «приемник занят», адаптер-источник повторяет передачу пакета. Отсутствие метки приемника в битах 36–37 воспринимается адаптером источником как метка «нет реакции приемника».

Для устранения некоторых ошибок функционирования сети используется станция-монитор. Монитор отмечает каждый проходящий пакет установкой бита 3 в единицу. Если поступает пакет с меткой монитора, равной 1, значит, система-источник по какой-то причине не освободила пакет. В этом случае очистку пакета выполняет монитор. При запуске сети монитор генерирует набор свободных пакетов, число которых фиксировано и зависит от протяженности кольца. При этом учитывается, что каждый адаптер по кольцу вносит задержку 2 бита и задержка кабеля составляет 4 бита на каждые 100 м длины.

В сети CR реализована передача данных в режиме сеанса связи, предусматривающая передачу серии взаимосвязанных пакетов, длина которой определяется при генерации сети. Для установления сеанса связи источник формирует и передает пакет, несущий заголовок сеанса. В заголовке указывается число пакетов; передаваемых в одном блоке. Если приемник готов принять блок данных, он возвращает источнику пакет с заголовком сеанса и меткой «пакет принят»; в ответ источник передает адресату последовательность пакетов, образующих блок данных. Блок снабжается контрольной суммой длиной 16 бит. Кроме того, в сети CR имеется узел управления именованьем, содержащий микропроцессор. Узел выполняет преобразование логических имен в физические адреса станций и обратное преобразование, благодаря чему абоненты сети могут пользоваться произвольной системой именованья, не связанной с физическими адресами узлов.

Проект 802 стандарта ЛВС. Институт инженеров по электротехнике и радиоэлектронике (IEEE) в США подготовил в 1981 г проект стандарта локальных сетей. Проект охватывает только два нижних уровня сетевых средств – управление физическим и информационным каналом, включая управление доступом к каналу, и устанавливает протоколы уровней 1, 2.1 и 2.2 (см. рис. 6.2).



Рис. 6.25. Формат кадра ЛВС (проект 802)

В основу управления информационным каналом положен протокол NDLC – стандартный протокол второго уровня, используемый в сетях передачи данных с коммутацией пакетов. Формат кадра, определяемый проектом 802, представлен на рис. 6.25. Кадр обрамлен флаговыми комбинациями 01111110, используемыми на уровне управления физическим каналом. Адреса назначения и источника кодируются одним (как и на рисунке) или несколькими байтами (до семи). Первый разряд байта адреса называется битом расширения адреса. Когда его значение равно 1, байт адреса является последним. Если бит расширения равен 0, адрес продолжается в следующем байте кадра. Например, двухбайтовый адрес назначения имеет следующую структуру:

0 L/G D D D D D D 1 D D D D D D D D

а двухбайтовый адрес источника –

0 C/R S S S S S S 1 S S S S S S S S

где *D* и *S* – биты адреса, принимающие значение 0 или 1.

Второй бит адреса источника *L/C* – идентификатор типа адреса: *L/G=0* обозначает индивидуальный адрес, а *L/G=1* – групповой адрес. Групповой адрес относится к совокупности систем. Единицы во всех битах адреса назначения определяют глобальный адрес, относящийся ко всем абонентам сети. Бит *C/R* адреса источника используется для обозначения команд и ответов: *C/R = 0* – команда; *C/R = 1* – ответ. Адрес источника может быть только индивидуальным. Байт управления используется для обозначения типа кадра.

Применяются три типа кадров: *I* – информационный, *C* – управляющий, *U* – нумерованный. Тип кадра определяется значением одного или двух первых битов поля управления. Информационные кадры служат для передачи нумерованных кадров и несут в поле управления следующие данные: *N(S)* – порядковый номер передаваемого кадра по модулю 8; *N(R)* – порядковый номер следующего принимаемого кадра по модулю 8; *P/F* – бит запроса – ответа. Кадр команды с битом запроса *P/F=1* используется для обращения к системе-адресату с целью получения кадра ответа, который должен идентифицироваться битом *P/F = 0*. Управляющие кадры *C* предназначены для передачи подтверждений о приеме кадров, запросов на повторную передачу информационных кадров и на временное прекращение передачи кадров *I*. В кадрах *C* поле *CC* указывает функцию управления, определяемую кадром. Ненумерованные кадры *U* используются для реализации дополнительных функций управления передачей данных и для передачи ненумерованных пакетов, данных. Область данных может содержать любое целое число байтов, а в управляющих и ненумерованных кадрах может отсутствовать. Код циклической суммы *КЦС* служит для проверки корректности кадра. Допускается использование 16-и 32-разрядных циклических сумм.

Состав команд и ответов, переносимых кадрами разных типов, перечислен в табл. 6.3. Кадр «Запрос» используется для повторной передачи кадров типа I , начиная от кадра с номером $N(R)$. Команда «Идентификация станции» служит для получения данных о типе системы, предельном размере принимаемого кадра и др. С помощью ответа «Отказ» система, выдавшая команду, извещается о переходе отвечающей системы в состояние, которое не может быть исправлено посредством повторной передачи.

Проект 802 определяет два способа доступа к моноканалу: свободный доступ с проверкой несущей и столкновений и эстафетный доступ. Способ СДПНС реализуется так же, как в сети Ethernet. Эстафетный доступ распространяется как на кольцевую, так и на магистральную структуру моноканала. В каждом случае эстафета передается последовательно от системы к системе. При этом в магистральной структуре организуется логическое кольцо за счет того, что система передает эстафету следующей системе в кадре управления, получает ответ о приеме кадра и переходит в состояние ожидания приема данных, в котором пребывает до получения эстафеты. Для поддержания работы кольца используются специальные процедуры контроля использования моноканала, изменения структуры кольца при отказе станций и создания кольца.

Таблица 6.3. Состав команд и ответов

Тип кадра	Команда	Ответ
I	Данные	Данные
C	Готов к приему Не готов к приему Запрос	Готов к приему Не готов к приему Повторная передача
U	Ненумерованные данные Разъединение Асинхронный сбалансированный режим Идентификация станции	Ненумерованное подтверждение Режим разъединения Отказ Идентификация станции

Проект 802 предусматривает четыре стандартные скорости передачи данных: 1; 5, 10 и 20 Мбит/с. В качестве передающей среды допускается использование экранированных скрученных пар, коаксиального или волоконно-оптического кабеля.

ОСНОВЫ ТЕОРИИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

7.1. ПРЕДМЕТ И ЗАДАЧИ

Теория вычислительных систем – инженерная дисциплина, объединяющая методы решения задач проектирования и эксплуатации ЭВМ, вычислительных комплексов, систем и сетей.

Предмет теории. Предметом исследования в теории вычислительных систем являются вычислительные системы в аспектах их производительности, надежности и стоимости. В системе выделяются следующие составляющие:

- 1) технические средства, определяемые конфигурацией системы – составом устройств и структурой связей между ними;
- 2) режим обработки, определяющий порядок функционирования системы;
- 3) рабочая нагрузка, характеризующая класс обрабатываемых задач и порядок их поступления в систему.

Когда ЭВМ, вычислительный комплекс, система или сеть исследуется в целом, как органическое единство составляющих во взаимодействии с окружающей средой, и при этом проявляются общесистемные свойства и характеристики, говорят, что исследование проводится на системном уровне. Представление исследуемых объектов (ЭВМ, комплексы, системы и сети) на системном уровне – наиболее характерная черта теории вычислительных систем.

Предметом исследования может быть функционирование процессора, внешнего запоминающего устройства и канала ввода – вывода, обмен данными между уровнями памяти, планирование, обработка, системный ввод – вывод и др. При этом свойства элементов и подсистем изучаются применительно к целям исследования всей системы, например к оценке производительности, и рассматриваются как части системы, функционирующие во взаимодействии с остальными частями.

Задачи анализа. Анализ вычислительных систем – определение свойств, присущих системе или классу систем. Типичная задача анализа – оценка производительности и надежности систем с заданной конфигурацией, режимом функционирования и рабочей нагрузкой. Другие примеры задач: определение (оценка) вероятности конфликта при доступе к общей шине, распределения длительности занятости процессора, загрузки канала ввода – вывода.

В общем случае задача анализа формулируется следующим образом. Исходя из цели исследования назначается набор характеристик $Y = \{y_1, \dots, y_M\}$ исследуемого объекта (вычислительная система, ее элемент, подсистема, некоторый процесс и др.) и точность $\Delta = \{\delta_1, \dots, \delta_M\}$, с которой они должны быть определены. Требуется найти способ оценки характеристик Y объекта с заданной точностью Δ и на основе этого способа определить характеристики.

При анализе систем в процессе эксплуатации оценка характеристик Y производится, как правило, измерением параметров функционирования с обработкой измерительных данных. В этом случае используется методика, устанавливающая состав измеряемых параметров, периодичность и длительность измерений, а также измерительные средства и средства обработки данных. В целях сокращения затрат на анализ стремятся измерять по возможности меньшее число наиболее просто измеримых параметров $X = \{x_1, \dots, x_N\}$, а

требуемый набор характеристик определять косвенным методом – вычислением с использованием зависимостей $y_m = \varphi_m(X), m = 1, \dots, M$. Эти зависимости либо имеют статистическую природу, либо создаются на основе фундаментальных закономерностей теории вычислительных систем.

При анализе проектируемых систем для оценки характеристик Y необходимо располагать моделью F , устанавливающей зависимость $Y = F(X)$ характеристик от параметров системы X , определяющих ее конфигурацию, режим функционирования, рабочую нагрузку. В этом случае решение задачи сводится к проведению на модели экспериментов, позволяющих дать ответы на интересующие вопросы. Точность оценки характеристик проектируемой системы зависит от адекватности модели и погрешности измерения параметров X .

Задачи идентификации. При эксплуатации вычислительных систем возникает необходимость в повышении их эффективности путем подбора конфигурации и режима функционирования, соответствующих классу решаемых задач и требованиям к качеству обслуживания пользователей. В связи с ростом нагрузки на систему и переходом на новую технологию обработки данных может потребоваться изменение конфигурации системы, использование более совершенных операционных систем и реализуемых ими режимов обработки. В этих случаях следует оценить возможный эффект, для чего необходимы модели производительности и надежности системы. Построение модели системы на основе априорных сведений об ее организации и данных измерений называется *идентификацией системы*.

Порядок идентификации вычислительной системы иллюстрируется рис. 7.1. В соответствии с природой исследуемых явлений для их представления предлагается функциональная модель, описывающая явления с точностью до значений параметров функций. Процесс создания такой модели называется *функциональной идентификацией* системы. В качестве функциональных моделей могут использоваться различные математические системы – дифференциальные и алгебраические уравнения, сети массового обслуживания и др., адекватно представляющие исследуемые аспекты.

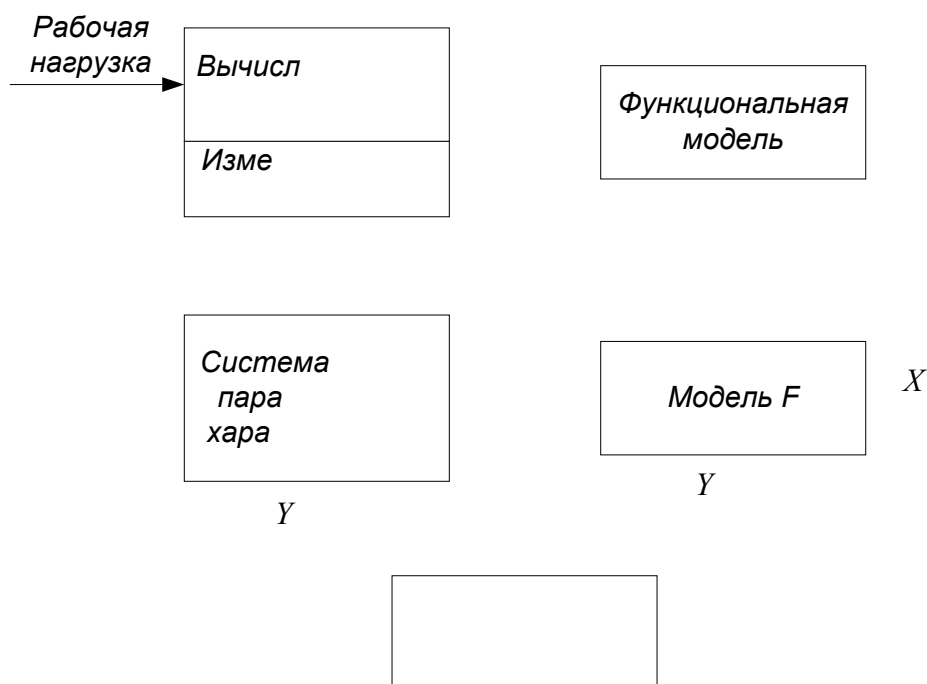


Рис. 7.1. Схема идентификации вычислительной системы

После того как выбрана функциональная модель, необходимо определить ее параметры. Этот процесс называется *параметрической идентификацией*. Для параметрической идентификации к вычислительной системе подключаются необходимые измерительные средства. Получаемые данные используются системой оценки параметров и характеристик для вычисления параметров X^* и характеристик Y^* системы, а также параметров модели $A=\{a_n\}$. Система оценки представляет собой набор программ для обработки измерительных данных, реализующий набор методов оценки параметров и характеристик. Вычисленные значения параметров A вводятся в модель, полностью определяя ее. Значения параметров X^* и характеристик Y^* системы используются для проверки адекватности модели, т.е. оценки погрешности Δ воспроизведения моделью характеристик системы. Оценка производится путем сравнения значений характеристик $Y=F(X^*)$, порождаемых моделью, с зарегистрированными характеристиками Y^* системы. Если модель адекватна системе, то используется для прогнозирования свойств системы, что сводится к вычислению на основе модели характеристик $Y=F(X)$, соответствующих новым значениям X параметров системы.

Задачи синтеза. Синтез – процесс создания вычислительной системы, наилучшим образом соответствующей своему назначению. Исходными в задаче синтеза являются следующие сведения, характеризующие назначение системы: 1) функция системы (класс решаемых задач); 2) ограничения на характеристики системы, например на производительность, время ответа, надежность и др.; 3) критерий эффективности, устанавливающий способ оценки качества системы в целом. Необходимо выбрать конфигурацию системы и режим обработки данных, удовлетворяющие заданным ограничениям и оптимальные по критерию эффективности. Типичная постановка задачи синтеза: спроектировать систему, обеспечивающую решение заданного класса задач A с производительностью не менее Λ задач в час, средней наработкой на отказ не менее T_0 и минимальной стоимостью.

Математически задача синтеза вычислительной системы формулируется следующим образом. Пусть $\Theta=(\theta_1, \dots, \theta_Q)$ – вектор параметров, характеризующих класс задач A , решение которых является функцией системы; $S=(s_1, \dots, s_P)$ – вектор параметров, характеризующий конфигурацию (структуру) системы; $C=(c_1, \dots, c_R)$ – вектор параметров режима обработки; $Y=(y_1, \dots, y_M)$ – вектор характеристик системы, связанный с параметрами задач θ , конфигурацией S и режимом обработки C зависимостью $Y = F(\Theta, S, C)$; $S = \{S_i\}$ – множество возможных конфигураций вычислительных систем; $C = \{C_j\}$ – множество возможных режимов обработки. Ограничения на характеристики и параметры системы $z_\alpha, \dots, z_\omega \in (YUX)$ будем представлять в виде $z_\alpha \in z_\alpha^*, \dots, z_\omega \in z_\omega^*$ где $z_\alpha^*, \dots, z_\omega^*$ – области допустимых значений соответствующих характеристик и параметров. Критерий эффективности системы представляется заданной функцией $E=\Phi(Y)$, зависящей от характеристик системы, которые в свою очередь предопределяются ее параметрами $Y = F(\Theta, S, C)$. В установленных обозначениях задача синтеза вычислительной системы формулируется так: определить конфигурацию S и режим обработки C , максимизирующие эффективность системы.

$$\max E = \max_{S \in s, C \in c} \Phi(Y) \quad (7.1)$$

при выполнении ограничений

$$z_\alpha \in z_\alpha^*, \dots, z_\omega \in z_\omega^* \quad (7.2)$$

В отличие от задачи анализа, направленной на определение характеристик системы Y по заданным параметрам X , задача синтеза состоит в определении параметров конфигурации S и режима обработки C , соответствующих параметрам рабочей нагрузки θ и характеристикам системы Y , заданным в виде (7.1), (7.2). Для задач синтеза характерны два следующих момента; Во-первых, предполагается наличие модели $Y = F(\theta, S, C)$, устанавливающей зависимость характеристик системы от ее параметров. Во-вторых, задача синтеза представляет собой оптимизационную задачу и предполагает использование метода оптимизации, соответствующего виду целевой функции (7.1) и ограничениям (7.2). Метод оптимизации должен гарантировать определение глобального оптимума целевой функции $E = \Phi(Y)$, определенной на множестве конфигураций S и режимов обработки C .

Сложность задачи синтеза вычислительной системы обусловлена числом варьируемых параметров, описывающих конфигурацию и режим функционирования системы, и областью варьирования параметров. При общей постановке задачи синтеза, когда множества конфигураций S и режимов обработки C включают в себя все мыслимые варианты построения систем (одномашинные и многомашинные, мультипроцессорные и сетевые) и различные способы управления задачами, данными и заданиями, сложность задачи синтеза превосходит возможности методов моделирования и оптимизации. Поэтому в общей постановке задача синтеза вычислительных систем оказывается неразрешимой. Для решения задачи синтеза ее упрощают разделением на последовательность этапов, на каждом из которых выявляются отдельные аспекты организации системы.

1. Исходя из назначения системы (класс решаемых задач, технология обработки данных, требования к производительности и условия работы) и состояния элементной базы определяется класс вычислительной системы: одномашинная система, мультипроцессорный комплекс, локальная сеть и др. При этом анализируется эффективность систем разных классов и выбирается класс, наилучшим образом удовлетворяющий назначению системы.

2. В выбранном классе систем синтезируется архитектура системы: определяется состав устройств, их функциональные возможности и технические характеристики, типы интерфейсов и структура связей между устройствами, наилучшим образом соответствующие назначению системы.

3. Определяется режим обработки данных и его параметры (состав и функции системных процессов, алгоритмы распределения ресурсов между заданиями и задачами, типы и диапазоны приоритетов и др.). Этим устанавливаются функции управляющих программ операционной системы и состав системного программного обеспечения.

Даже при разделении задачи синтеза на три или большее число этапов синтез, связанный с каждым этапом не удастся свести к единой математической процедуре – задаче математического программирования. Это обусловлено двумя основными причинами. Во-первых, синтез связан с разнотипными параметрами: одни являются количественными, а другие – качественными, т. е. признаками типа структуры, устройств, памяти, интерфейсов, способов управления процессами и др. Поэтому синтез сводится к задачам численного математического программирования, а также к комбинаторным задачам на сочетаниях, отношениях и т. д. Объединение разнотипных задач в одну задачу оптимизации оказывается нерезультативным из-за разнотипности вычислительных процедур, которые должны использоваться в процессе оптимизации. Во-вторых, имеющиеся в распоряжении исследователей модели носят, как правило, локальный характер, воспроизводя свойства достаточно узкого класса структурных решений и режимов функционирования. Объединение простых моделей в сложную приводит к разрывам функций, невыпуклым зависимостям, что не только затрудняет, но практически исключает возможность применения методов оптимизации.

По этим причинам при синтезе систем стремятся по возможности уменьшать размерность задач путем разделения задачи синтеза на последовательность этапов, сводящихся к чисто комбинаторным задачам или задачам численной оптимизации. При этом проектирование системы ведется сверху вниз – от наиболее общих решений, связанных с

системой в целом, к частным решениям, относящимся к отдельным подсистемам и их частям. При решении многих задач синтеза приходится использовать весьма простые модели функционирования системы и ее составляющих, приближенно представляющие зависимости между характеристиками и параметрами. В этих условиях выбор проектных решений производится на основе опыта и интуиции разработчиков. Таким образом, синтез вычислительных систем сводится к решению значительного числа взаимосвязанных задач выбора способов организации и определения параметров проектируемой системы в различных аспектах ее организации и в отношении к различным подсистемам и элементам. При этом используются как формальные, так и эвристические методы, причем на долю последних приходится значительное число проектных задач, выходящих за рамки возможностей известных методов теории вычислительных систем.

7.2. МОДЕЛИ И МЕТОДЫ

Специфика предмета и задач теории вычислительных систем порождает особый класс моделей, используемых для представления порядка функционирования систем и их составляющих с целью прогнозирования производительности, надежности и других характеристик, а также совокупность методов решения задач анализа, идентификации и синтеза, опирающихся на соответствующие модели. Задачи анализа производительности и надежности, а также задачи синтеза систем с заданной производительностью и надежностью – наиболее массовые задачи проектирования и эксплуатации вычислительных систем. Поэтому в теории вычислительных систем наибольшее внимание уделяется моделям производительности и надежности и методам обеспечения требуемой производительности и надежности при проектировании и эксплуатации систем различного назначения.

Принципы построения и свойства моделей. *Модель* – физическая или абстрактная система, адекватно представляющая объект исследования. В теории вычислительных систем используются преимущественно *абстрактные модели* – описания объекта исследования на некотором языке. Абстрактность модели проявляется в том, что компонентами модели являются не физические элементы, а понятия, в качестве которых наиболее широко используются математические. Абстрактная модель, представленная на языке математических отношений, называется *математической моделью*. Математическая модель имеет форму функциональной зависимости $Y=F(X)$, где $Y = \{y_1, \dots, y_M\}$ и $X = \{x_1, \dots, x_N\}$ – соответственно характеристики и параметры моделируемой системы и F – функция, воспроизводимая моделью. Построение модели сводится к выявлению функции F и представлению ее в форме, пригодной для вычисления значений $Y=F(X)$. Модель позволяет оценивать характеристики Y для заданных параметров X и выбирать значения параметров, обеспечивающие требуемые характеристики, с использованием процедур оптимизации.

Модель создается исходя из цели исследования, устанавливающей:

- 1) состав воспроизводимых характеристик $Y = \{y_1, \dots, y_M\}$,
- 2) состав параметров $X = \{x_1, \dots, x_N\}$, изменение которых должно влиять на характеристики Y ;
- 3) область изменения параметров $x_n \in x_n^*, n = 1, \dots, N$, – область определения модели.
- 4) точность – предельная допустимая погрешность оценки характеристик Y на основе модели.

Состав характеристик Y определяется в зависимости от исследуемых свойств системы – производительности, надежности и других и должен гарантировать полноту отображения этих свойств. Состав параметров X должен охватывать все существенные аспекты организации системы, изучение влияния которых на качество функционирования составляет цель исследования, производимого с помощью модели. Область определения модели характеризует диапазон исследуемых вариантов организации систем. Чем обширнее состав характеристик и параметров, а также область определения модели, тем универсальнее

модель в отношении задач, которые можно решать с ее использованием. Предельные допустимые погрешности оценки характеристик и точность задания параметров определяют требования к точности модели. Так, если изменения характеристик в пределах 10% несущественны для выбора того или другого варианта построения системы, то точность определения характеристик должна составлять $\pm 5\%$. В большинстве случаев параметры, в первую очередь параметры рабочей нагрузки, могут быть заданы лишь приблизительно, с относительной погрешностью 10–25%. В таких случаях нет смысла предъявлять высокие требования к точности воспроизведения моделью характеристик системы и погрешности их оценки на уровне 5–15 % вполне приемлемы.

Модель, удовлетворяющая вышеперечисленным требованиям по составу характеристик и параметров и точности воспроизведения характеристик во всей области определения, называется *адекватной* системе. Свойство адекватности модели является относительным, связанным с целью исследования. Больше всего адекватность проявляется в точности воспроизведения характеристик системы моделью. Существенное влияние на адекватность оказывает область определения модели. Практически любая модель обеспечивает высокую точность воспроизведения характеристик в пределах малой окрестности точки $X = \{x_1, \dots, x_N\}$, но только высококачественные модели гарантируют точность характеристик в широком диапазоне параметров X . Чем шире область определения модели, тем меньше шансов, что некоторая модель окажется адекватной системе.

Другое свойство модели – *сложность*. Сложность модели принято характеризовать двумя показателями: размерностью и сложностью вычислений, связанных с определением характеристик. Размерность модели – число величин, представляющих в модели параметры и характеристики. Так, если модель F_A служит для вычисления двух характеристик, зависящих от 5 параметров, а модель F_B – двух характеристик, зависящих от 10 параметров, то размерность модели F_A равна 7, а модели F_B – 12 и модель F_B рассматривается как более сложная. Сложность вычислений, выполняемых при расчете характеристик $Y=F(X)$, оценивается числом операций, приходящихся на одну реализацию оператора P . Обычно сложность вычислений связывается с затратами ресурсов ЭВМ и характеризуется числом процессорных операций и емкостью памяти для хранения информации, относящейся к модели. Сложность вычислений – монотонно возрастающая функция размерности модели. Поэтому более сложной модели присущи одновременно большая размерность и сложность вычислений.

Сложность модели определяется сложностью моделируемой системы и назначением модели (состав характеристик и параметров, воспроизводимых в модели), размером области определения и точностью модели. Чем сложнее система, т. е. чем больше число входящих в нее элементов и процессов, из которых складывается функционирование системы, тем сложнее модель. Увеличение числа воспроизводимых характеристик и параметров, области определения и точности оценки характеристик приводят к увеличению сложности модели.

Вероятностный подход к моделированию процессов. Производительность и надежность вычислительных систем связаны с временными аспектами функционирования. При оценке производительности первостепенное значение имеет продолжительность вычислительных процессов. При оценке надежности исследуется продолжительность пребывания системы в различных состояниях, которые меняются из-за отказов оборудования и последующего восстановления работоспособности. Для вычислительных систем, рассматриваемых на системном уровне, типично наличие случайных факторов, влияющих на характер протекания процессов. Так, продолжительность процессорной обработки, число и порядок обращений к периферийным устройствам зависят от исходных данных, которые порождаются вне системы и носят для нее случайный характер. Случайными являются поток отказов и время восстановления отказавших элементов. В связи с этим при оценке функционирования вычислительных систем используется вероятностный подход, предполагающий, что на процессы воздействуют случайные факторы и свойства процессов проявляются статистически, на множестве их реализаций.

Процессы, происходящие в вычислительных системах, представляются в моделях как непрерывные или дискретные случайные процессы. При исследовании вычислительных систем чаще всего приходится иметь дело с дискретными случайными процессами определенными на конечном множестве состояний, причем процессы рассматриваются или в непрерывном, или в дискретном времени.

Вероятностный подход к описанию функционирования вычислительных систем приводит к использованию аппарата теории вероятностей и математической статистики в качестве математической базы методов исследования.

Случайные величины, соответствующие параметрам, характеристикам и другим элементам моделей, могут представляться на разных уровнях, среди которых наиболее широко используются следующие четыре: 1) статистическая выборка a_1, \dots, a_n , определяющая случайную величину набором значений, имевших место в некоторой реализации случайного процесса; 2) закон распределения случайной величины; 3) математическое ожидание и дисперсия; 4) математическое ожидание. На первом уровне случайная величина определяется наиболее полно, с наибольшей подробностью, а на последнем уровне – наименее детально.

Марковские модели. Основополагающими в теории вычислительных систем являются модели и аппарат теории марковских процессов. *Марковским* называется случайный процесс, состояние которого в очередной момент времени $t + \delta$ зависит только от текущего состояния в момент времени t . Это означает, что поведение марковского процесса в будущем определяется текущим состоянием процесса и не зависит от предыстории процесса – состояний, в которых пребывал процесс до момента t .

В классе марковских процессов выделяют процессы с дискретными состояниями, называемые *марковскими цепями*. Когда множество состояний процесса $S = \{s_1, \dots, s_K\}$ конечно марковскую цепь называют *конечной*. Конечная марковская цепь может быть определена в непрерывном или дискретном времени ч. В первом случае переходы процесса из одного состояния в другое связываются с произвольными моментами времени t_0, t_1, t_2, \dots и цепь называют *непрерывной*; во втором – только в фиксированные моменты времени, обозначаемые порядковыми номерами $t = 0, 1, 2, \dots$, и цепь называется *дискретной*.

Дискретная марковская цепь определяется:

1) множеством состояний $S = \{s_1, \dots, s_K\}$

2) матрицей вероятностей переходов (переходных вероятностей) характеризующей вероятности перехода процесса с текущим состоянием s_i в следующее состояние s_j ;

$$P = [p_{ij}] = \begin{matrix} & \begin{matrix} s_1 & s_2 & \dots & s_K \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ \dots \\ s_K \end{matrix} & \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1K} \\ p_{21} & p_{22} & \dots & p_{2K} \\ \dots & \dots & \dots & \dots \\ p_{K1} & p_{K2} & \dots & p_{KK} \end{bmatrix} \end{matrix} \quad (7.3)$$

3) вектором начальных вероятностей (начальным распределением) $\pi_0 = \{p_1^{(0)}, \dots, p_K^{(0)}\}$ определяющим вероятности $p_i^{(0)}$ того, что в начальный момент времени $t=0$ процесс находится в состоянии S_i .

Марковская цепь изображается в виде графа, вершины которого соответствуют состояниям цепи и дуги – переходам между состояниями. Дуги (i, j) , связывающие вершины s_i и s_j , отличаются вероятностями переходов p_{ij} . На рис. 7.2 представлен граф марковской цепи с множеством состояний $S = \{s_1, \dots, s_5\}$, матрицей вероятностей переходов

$$P = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 & s_4 & s_5 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0,4 & 0,3 & 0,2 & 0,1 \\ 0 & 1 & 0 & 0 & 0 \\ 0,1 & 0,9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

и вектором начальных вероятностей $\pi_0 = \{1, 0, 0, 0, 0\}$.

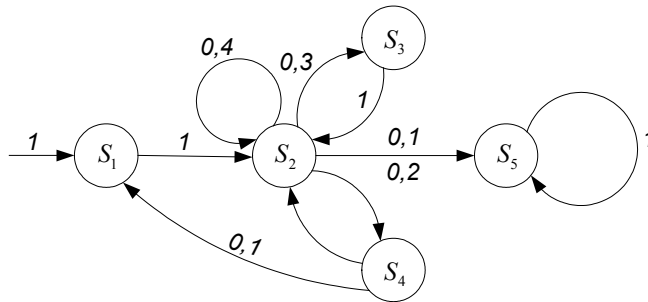


Рис. 7.2. Граф марковской цепи

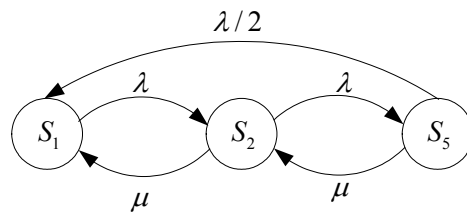


Рис. 7.3. Граф непрерывной марковской цепи

Марковская цепь порождает множество реализаций случайного процесса $f(t)$, который представляется последовательностью состояний $f(t) = f_0, f_1, f_2, \dots, f(t) \in S$, соответствующих моментам времени $t=0, 1, 2, \dots$. Начальное состояние $f_0 = s_i$ определяется вектором начальных вероятностей π . Следующее состояние $f_1 = s_j$ определяется i -й строкой матрицы вероятностей переходов P : процесс $f(t)$ переходит в состояние $f_1 = s_j$ с вероятностью p_{ij} . Затем процесс переходит в состояние $f_2 = s_k$, определяемое вероятностями p_{jk} , соответствующими состоянию S_j , и т. д. В результате n шагов процесс попадает в состояния s_1, \dots, s_K с вероятностями $\pi_n = \{p_1^{(n)}, \dots, p_K^{(n)}\}$ соответственно.

Марковские цепи классифицируются в зависимости от возможности перехода из одних состояний в другие. Основными являются два класса: поглощающие и эргодические цепи.

Поглощающая марковская цепь содержит поглощающее состояние, достигнув которого, процесс уже никогда его не покидает, т. е. по сути прекращается. Поглощающее состояние будем обозначать s_0 . Вероятность перехода $p_{00} = 1$ и, следовательно, все остальные вероятности $p_{0j} = 0, j = 1, \dots, K$. Матрица вероятностей переходов поглощающей цепи имеет следующий вид:

$$P = [p_{ij}] = \begin{matrix} & \begin{matrix} s_0 & s_1 & \dots & s_K \end{matrix} \\ \begin{matrix} s_0 \\ s_1 \\ \dots \\ s_K \end{matrix} & \begin{bmatrix} 1 & 0 & \dots & 0 \\ p_{10} & p_{11} & \dots & p_{1K} \\ \dots & \dots & \dots & \dots \\ p_{K0} & p_{K1} & \dots & p_{KK} \end{bmatrix} \end{matrix} \quad (7.4)$$

Из какого бы состояния ни начался процесс, при $n \rightarrow \infty$ с вероятностью 1 он окажется в поглощающем состоянии s_0 . Основная характеристика случайного процесса, порождаемого поглощающей марковской цепью, – число пребываний процесса в состояниях s_1, \dots, s_K до момента поглощения. Число пребываний в каждом из состояний $S_i, i=1, \dots, K$ и на множестве невозвратных состояний $\{s_1, \dots, s_K\}$ – случайные величины, характеризуемые средними значениями, дисперсиями и распределениями. Для определения указанных характеристик используются методы алгебраической теории марковских цепей [14].

Поглощающие марковские цепи широко используются в качестве временных моделей программ и вычислительных процессов. При моделировании программы состояния цепи отождествляются с блоками программы, а матрица переходных вероятностей определяет порядок переходов между блоками, зависящий от структуры программы и распределения исходных данных, значения которых влияют на развитие вычислительного процесса. В результате представления программы поглощающей цепью удается вычислить число обращений к блокам программы и время выполнения программы, оцениваемое средними значениями, дисперсиями и при необходимости – распределениями. Аналогично вычислительный процесс, сводящийся к последовательности обращений к ресурсам системы в порядке, определяемом программой, можно представить поглощающей марковской цепью, состояния которой соответствуют использованию ресурсов системы – процессора и периферийных устройств, а переходные вероятности отображают порядок обращения к различным ресурсам. За счет этого вычислительный процесс представляется в компактной форме, удобной для анализа характеристик процесса.

Эргодическая марковская цепь представляет собой множество состояний, связанных матрицей переходных вероятностей таким образом, что из какого бы состояния процесс ни исходил, после некоторого числа шагов он может оказаться в любом состоянии. Это означает, что в любое состояние эргодической цепи можно перейти из любого другого состояния за сколько-то шагов. По этой причине состояния эргодической цепи называются эргодическими (возвратными). Процесс, порождаемый эргодической цепью, начавшись в некотором состоянии, никогда не завершается, а последовательно переходит из одного состояния в другое, попадая в различные состояния с разной частотой, зависящей от переходных вероятностей. Поэтому основная характеристика эргодической цепи – вероятности пребывания процесса в состояниях $S_j, j=1, \dots, K$, – относительные частоты попадания процесса в состояния S_j и одновременно доля времени, которую процесс проводит в каждом из состояний. В качестве дополнительных характеристик, эргодических цепей используются математическое ожидание и дисперсия времени (числа шагов) первого попадания в состояние S_j из состояния S_i и предельная корреляция числа попаданий в состояния S_i и S_j . Эти характеристики определяются методами алгебраической теории марковских цепей [8].

Эргодические цепи широко используются в качестве моделей надежности систем. При этом состояния системы, различающиеся составом исправного и отказавшего оборудования, трактуются как состояния эргодической цепи, переходы между которыми связаны с отказами и восстановлением устройств и реконfigurацией связей между ними, проводимой для сохранения работоспособности системы. Оценки характеристик эргодической цепи дают представление о надежности поведения системы в целом. Кроме

того, эргодические цепи широко, используются в качестве базовых моделей взаимодействия устройств с задачами, поступающими на обработку.

Марковский процесс с дискретными состояниями s_1, \dots, s_K , переходы между которыми разрешаются в любой момент времени, называется *непрерывной марковской цепью*. Однородная непрерывная марковская цепь, поведение которой в любой момент времени подчиняется одному и тому же закону, задается матрицей интенсивностей переходов $Q = [q_{ij}]$, $i, j = 1, \dots, K$. Интенсивность переходов определяется следующим образом:

$$q_{ij} = \lim_{\Delta t \rightarrow 0} \frac{p_{ij}(\Delta t) - 1}{\Delta t}; \quad q_{ij} = \lim_{\Delta t \rightarrow 0} \frac{p_{ij}(\Delta t)}{\Delta t}$$

где $p_{ii}(\Delta t)$ – вероятность перехода процесса из состояния s_i в состояние s_i , за время Δt .

Это означает, что если процесс находится в состоянии S_i , то вероятность перехода в течение промежутка времени Δt в состояние S_j , отличное от S_i , равна $-q_{ii}\Delta t$. Аналогично вероятность перехода процесса в течение промежутка времени Δt из состояния S_i в состояние S_j равна $q_{ij}\Delta t$. Интенсивность переходов должна удовлетворять условию

$$\sum_{j=1}^K q_{ij} = 0, \quad i = 1, \dots, K \quad (7.5)$$

На рис. 7.3 представлен граф непрерывной марковской цепи с тремя состояниями S_1, S_2, S_3 . Дуги графа нагружены интенсивностями переходов. Графу соответствует следующая матрица интенсивностей переходов:

$$Q = [q_{ij}] = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{bmatrix} -\frac{3}{2}\lambda & \lambda & \lambda/2 \\ \mu & -(\mu + \lambda) & \lambda \\ 0 & \mu & -\mu \end{bmatrix} \end{matrix} \quad (7.6)$$

При построении матрицы значения q_{ii} , $i = 1, \dots, K$, в соответствии с (7.5) определяются следующим образом:

$$q_{ii} = -\sum_{\substack{j=1 \\ (j \neq i)}}^K q_{ij}$$

Основная характеристика непрерывной марковской цепи – стационарное (финальное) распределение вероятностей состояний $\alpha = \{a_1, \dots, a_K\}$, где a_1, \dots, a_K – вероятности пребывания процесса в состояниях s_1, \dots, s_K соответственно. Распределение задается вероятностным решением системы линейных уравнений

$$\alpha Q = 0 \quad (7.7)$$

которая в развернутой форме имеет следующий вид:

$$\sum_{j=1}^K q_{ji} a_j = 0, i = 1, \dots, K$$

Решение системы, составленной из $(K-1)$ уравнений (7.8) и нормирующего уравнения $a_1 + \dots + a_K = 1$, определяет значения вероятностей $\alpha = \{a_1, \dots, a_K\}$. Уравнения (7.7), (7.8) называются *уравнениями равновесия*. Они легко составляются по графу марковской цепи с учетом того, что в каждом состоянии входящий поток должен равняться исходящему потоку. Так, для цепи на рис. 7.3 имеем

Состояние	Интенсивность входящего потока	Интенсивность исходящего потока
s_1	$\mu\alpha_2$	$\left(\frac{\lambda}{2} + \lambda\right)a_1$
s_2	$\lambda\alpha_1 + \mu\alpha_3$	$(\mu + \lambda)a_2$
s_3	$\frac{\lambda}{2}a_1 + \lambda\alpha_2$	$\mu\alpha_3$

С учетом равенства интенсивности входящего и исходящего потока

$$\begin{cases} \mu\alpha_2 = (\lambda/2 + \lambda)a_1; \\ \lambda\alpha_1 + \mu\alpha_3 = (\mu + \lambda)a_2; \\ \frac{\lambda}{2}a_1 + \lambda\alpha_2 = \mu\alpha_3; \end{cases}$$

Полученная система является системой уравнений равновесия для цепи, изображенной на рис. 7.3 и заданной матрицей (7.6).

В соответствии с марковским свойством вся предыстория процесса сказывается на его поведении в будущем только через текущее состояние, которое и определяет дальнейший ход процесса. Таким образом, нет необходимости знать, как долго процесс находится в текущем состоянии. Отсюда следует, что распределение остающегося времени пребывания процесса в состоянии s_j должно зависеть только от самого состояния, а не от времени пребывания в нем. Этим свойством обладает только одно распределение – экспоненциальное, функция плотности вероятности которого имеет следующий вид: $p(t) = 1/\tau \exp(-t/\tau)$, где τ – параметр распределения, определяющий математическое ожидание случайной величины t . Таким образом, неперенное свойство непрерывного марковского процесса – экспоненциальность распределения времени пребывания процесса в каждом из состояний.

Если снять указанное ограничение на время пребывания процесса в состояниях s_1, \dots, s_K , т.е. допустить произвольное распределение времени пребывания, то процесс становится полумарковским. Полумарковский процесс ведет себя относительно моментов изменения состояний как обычная дискретная марковская цепь, и принято говорить, что в эти моменты времени имеет место вложенная марковская цепь. Характеристики полумарковского процесса определяются значительно сложнее, чем для марковского процесса, поскольку вероятности состояний связаны с параметрами процесса системой дифференциальных уравнений в частных производных. Поэтому вычисление стационарных вероятностей состояний в общем случае выливаются в сложную математическую задачу.

Модели массового обслуживания. Для построения моделей производительности вычислительных систем широко используется аппарат теории массового обслуживания. В ней изучаются системы, на вход которых поступает поток заявок (требований), приходящихся в общем случае в случайные моменты времени. Поступившая заявка

обслуживается в системе путем предоставления ей некоторых ресурсов на какое-то время и, будучи обслуженной, покидает систему. Определение времени пребывания заявок в системе составляет сущность теории массового обслуживания. Наиболее характерный момент функционирования систем массового обслуживания – наличие очередей, в которых поступившие заявки ждут момента освобождения ресурсов, занятых обслуживанием других, например, ранее поступивших заявок. ЭВМ можно рассматривать как систему массового обслуживания, на вход которой поступают задания, обслуживаемые путем предоставления им процессорного времени и времени остальных устройств ЭВМ. Анализ ЭВМ как системы массового обслуживания позволяет определить число заданий, находящихся в ЭВМ на различных стадиях обслуживания, время ожидания заданий в очередях, время пребывания задания в ЭВМ и другие характеристики процесса обработки заданий, функционирования ЭВМ и ее устройств.

Система массового обслуживания (рис. 7.4) состоит из входящего потока заявок a , очереди Q , дисциплины обслуживания D , определяющей порядок выбора заявок из очереди, и обслуживающего прибора Π или K одинаковых обслуживающих приборов (каналов) Π_1, \dots, Π_k . Система, содержащая только один прибор (канал), называется *одноканальной*, а несколько приборов, – *многоканальной*. Функционирование системы состоит в постановке поступающих заявок в очередь на обслуживание, выборе из очереди заявки, подлежащей первоочередному обслуживанию, и предоставлении прибора заявке на определенное время. По окончании обслуживания заявка покидает систему. На выходе системы образуется выходной поток заявок.

Таким образом, система массового обслуживания характеризуется следующим набором параметров:

- 1) распределением длительности интервалов между заявками входящего потока $p(a)$;
- 2) дисциплиной обслуживания заявок D ;
- 3) числом обслуживающих приборов (каналов) K ;
- 4) распределением длительности обслуживания заявок приборами (каналами) $p(b)$.

Указанный набор параметров полностью определяет порядок функционирования системы. Процесс функционирования количественно оценивается следующим набором основных характеристик:

- 1) загрузкой – средним по времени числом приборов (каналов), занятых обслуживанием (для одноканальной системы загрузка определяет долю времени, в течение которой прибор занят обслуживанием, т. е. не простаивает);
- 2) длиной очереди – числом заявок, ожидающих обслуживания;
- 3) числом заявок, находящихся в системе (в очереди и на обслуживании приборами);
- 4) временем ожидания заявки – от момента поступления заявки в систему до начала обслуживания;
- 5) временем пребывания заявки в системе – от момента поступления заявки до окончания ее обслуживания, т. е. до выхода из системы.

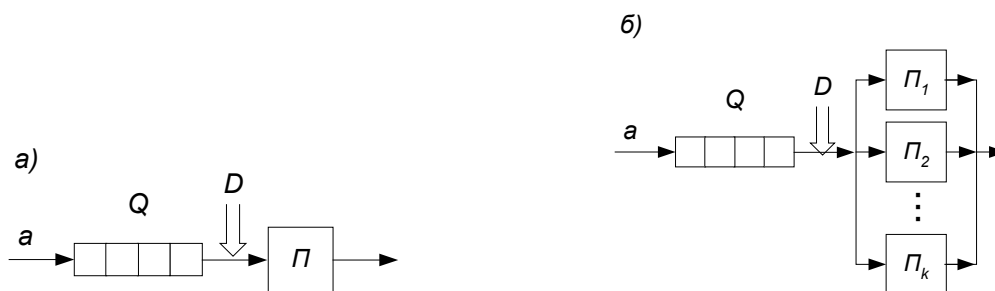


Рис. 7.4. Одноканальная (а) и многоканальная (б) система массового обслуживания

Наряду с основными характеристиками для оценки функционирования системы используются дополнительные характеристики: длительность простоя, непрерывной занятости приборов и др.

Все указанные характеристики, кроме загрузки, – случайные величины, представляемые соответствующими распределениями: распределением длины очереди, числа заявок в системе, времени ожидания и т. д. На более низком уровне детализации случайные величины характеризуются средними значениями и дисперсиями или только средними. Зависимости характеристик системы от ее параметров – предмет исследования элементарной теории массового обслуживания.

В теории массового обслуживания изучаются и более сложные объекты – сети. *Сеть массового обслуживания* – совокупность взаимосвязанных систем массового обслуживания. Пример сети представлен на рис. 7.5. Здесь S_0 – узел – источник заявок, S_1, \dots, S_4 – системы массового обслуживания и S_5 – узел, представляющий выход из сети. Дуги показывают направления движения заявок по сети. Если из вершины выходит единственная дуга, то все заявки, обслуженные соответствующей системой, направляются по этой дуге. Если из вершины выходит несколько дуг, то каждая заявка направляется по одной выходящей дуге, выбор которой производится в соответствии с вероятностями передачи заявок $p_{ij}, p_{ik}, \dots, p_{il}$ из узла S_i в узлы S_j, S_k, \dots, S_l соответственно, причем $p_{ij} + p_{ik} + \dots + p_{il} = 1$.

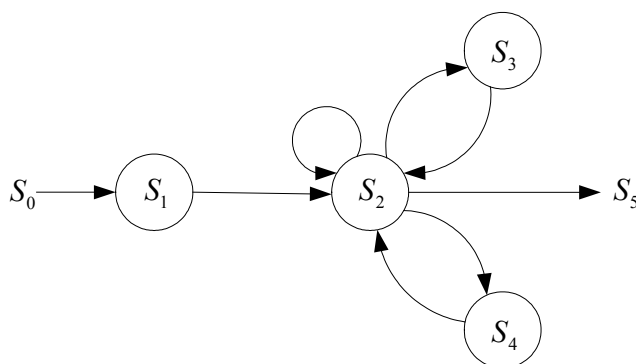


Рис. 7.5. Граф сети массового обслуживания

Сеть функционирует следующим образом. Заявка от источника S_0 поступает на обслуживание в систему, определенную дугой, выходящей из источника. После обслуживания в этой системе заявка поступает в следующую систему и циркулирует по сети, последовательно обслуживаясь в различных системах, до тех пор, пока не покинет сеть по дуге, ведущей к выходу. В сети заявка либо стоит в очереди к одной из систем, либо обслуживается прибором системы.

Сеть массового обслуживания задается следующим набором параметров:

- 1) параметрами источника заявок;
- 2) структурой, определяющей конфигурацию связей и вероятности передачи заявок между узлами сети;
- 3) параметрами систем массового обслуживания $S_i, i=1, \dots, N$ – дисциплиной обслуживания D_i , числом каналов K_i и распределением длительности обслуживания заявок $p(b_i)$.

Функционирование сети массового обслуживания определяется совокупностью узловых и сетевых характеристик. Узловые характеристики оценивают функционирование каждой системы массового обслуживания и включают в себя характеристики потока заявок, поступающего на вход узла, и весь набор характеристик, присущий системам массового

обслуживания. Сетевые характеристики оценивают функционирование сети в целом и включают в себя:

- 1) загрузку – среднее по времени число заявок, обслуживаемых сетью, и одновременно среднее число приборов (каналов), занятых обслуживанием;
- 2) число заявок, ожидающих обслуживания в сети;
- 3) число заявок, находящихся в сети (в состоянии, ожидания и обслуживания);
- 4) суммарное время ожидания заявки в сети;
- 5) суммарное время пребывания заявки в сети.

Теория массового обслуживания предлагает способы расчета характеристик сетей различных типов, а также способы выбора параметров сетей, обеспечивающих заданные характеристики функционирования.

Для воспроизведения в моделях различных способов организации процессов функциональные возможности сетей массового обслуживания расширяются путем включения в сети специальных узлов. Так, для отображения эффектов, связанных с использованием запоминающих устройств, в сетевые модели включаются узлы, моделирующие работу запоминающих устройств, – узлы памяти. Память характеризуется емкостью, которая распределяется между заявками. Обслуживание заявки, поступившей на вход узла памяти, сводится к выделению затребованного числа ячеек памяти. Если в памяти, отсутствует область требуемого размера, заявка ставится в очередь и ожидает момента освобождения памяти, предоставленной ранее поступившим заявкам. Возможности сети могут расширяться за счет использования специальных узлов, управляющих маршрутами заявок: направляющих заявку одновременно по нескольким маршрутам; синхронизирующих движение заявок; изменяющих атрибуты заявок и т. д. Сети, воспроизводящие процессы массового обслуживания в форме взаимодействия систем массового обслуживания и дополнительных узлов, моделирующих работу памяти (накопителей), источников и приемников заявок и процессы маршрутизации заявок, называются *стохастическими сетями*. Стохастические сети включают в себя сети массового обслуживания как один из вариантов организации процессов массового обслуживания.

В отличие от систем массового обслуживания, стохастические сети, в том числе и сети массового обслуживания, воспроизводят процессы многоэтапного обслуживания, когда обслуживание заявки производится за счет последовательного обращения к ресурсам, в том числе и многократного. Характерное свойство сети – ее структурное подобие реальной системе. Состав узлов сети l конфигурация связей между ними соответствует составу устройств и порядку их взаимодействия в реальной системе. За счет этого значительно упрощается процесс построения сетевых моделей и обеспечивается адекватность процессов функционирования сетей и моделируемых ими систем.

Статистические модели. В тех случаях, когда причинно-следственные отношения в исследуемом объекте трудно охарактеризовать из-за их многообразия, сложности и невыясненной природы процессов или когда эти отношения несущественны, а желательно представить свойства объекта в достаточно компактной форме, используются статистические методы для математического выражения зависимостей между характеристиками и параметрами объекта. *Статистические методы* – совокупность способов сбора, анализа и интерпретации данных о некотором объекте или совокупности объектов с целью получения теоретических или практических выводов.

Сущность статистических методов состоит в следующем. На основе эмпирических представлений о свойствах, исследуемого объекта и в соответствии с целью исследования определяется состав признаков, характеризующих объект, и тип статистической модели (математические выражения, структуры). Признаки, посредством которых описывается объект, – величины, соответствующие параметрам x_1, \dots, x_N и характеристикам y_1, \dots, y_M объекта. Наблюдением (измерения, регистрация) собираются статистические данные, образующие выборку следующего вида:

Номер наблюдения	x_1	...	x_N	y_1	...	y_M
1	$x_1^{(1)}$...	$x_N^{(1)}$	$y_1^{(1)}$...	$y_M^{(1)}$
2	$x_1^{(2)}$...	$x_N^{(2)}$	$y_1^{(2)}$...	$y_M^{(2)}$
...
n	$x_1^{(n)}$...	$x_N^{(n)}$	$y_1^{(n)}$...	$y_M^{(n)}$

где $x_1^{(i)}, \dots, x_N^{(i)}, y_1^{(i)}, \dots, y_M^{(i)}$ – значения признаков при i -м наблюдении.

На основе этой выборки строится статистическая модель заданного типа, устанавливающая количественную взаимосвязь признаков.

Математическая статистика предлагает обширный набор методов и методов установления статистических закономерностей, присущих исследуемым объектам. Наиболее широкое применение при исследовании вычислительных систем получил регрессионный анализ.

Регрессионный анализ состоит, в построении функций $y_i = f_i(x_1, \dots, x_N)$ связывающих характеристики (зависимые переменные) с параметрами (независимыми переменными), на основе статистической выборки, содержащей статистически независимые данные. Статистическая независимость данных состоит в том, что значения признаков разных наблюдений статистической выборки не должны зависеть друг от друга. Чтобы проявились статистические зависимости, число наблюдений должно превосходить число признаков в 6-8 раз. Выборка должна быть однородной, т. е. относиться к объектам одного класса.

Зависимость характеристики от параметров x_1, \dots, x_N представляется в виде линейного полинома

$$y = b_0 + b_1x_1 + \dots + b_Nx_N \quad (7.9)$$

а при необходимости – в виде полинома более высокого порядка

$$y = b_0 + b_1x_1 + \dots + b_Nx_N + b_{12}x_1x_2 + b_{13}x_1x_3 + \dots + b_{1N}x_1x_N + \dots + b_{123}x_1x_2x_3 + \dots + b_{11}x_1^2 + \dots$$

Параметры b называются *коэффициентами регрессии*. Если число признаков $N=1$, то (7.9) называют *уравнением парной регрессии*, если $N \geq 2$, – *уравнением множественной регрессии*. Переменная y рассматривается как случайная величина, которая распределена в окрестности среднего значения \bar{y} , зависящего от x_j т. е. считается, что переменные x_j влияют лишь на среднее значение \bar{y}

дисперсии отклонения уравнения регрессии от наблюдаемых значений $y^{(i)}, i = 1, \dots, n$.

При построении регрессионной модели основными являются два момента: 1) выбор числа независимых признаков x_1, \dots, x_N ; 2) выбор формы полинома, посредством которого представляется зависимость $y = f(x_1, \dots, x_N)$. Процедуры оценки качества и совершенствования моделей реализованы в пакетах прикладных программ статистического анализа, используемых при исследованиях.

Регрессионные модели обладают следующими особенностями. Во-первых, они применимы для прогноза значений y только при аргументах x_1, \dots, x_N , принадлежащих области определения переменных, для которой построено уравнение регрессии. Во-вторых, уравнения регрессии принципиально необратимы, т. е. недопустимо путем тождественных преобразований из уравнения $y = f(x_1, \dots, x_N)$ строить уравнение $x_j = \varphi(x_1, \dots, x_N, y)$, поскольку это две совершенно различные регрессии, каждая из которых должна строиться самостоятельно. Дополнительно отметим, что

регрессионные модели не раскрывают механизм взаимосвязи характеристик и параметров и фиксируют лишь количественную взаимосвязь величин.

Регрессионные и другие статистические модели наиболее широко используются для описания рабочей нагрузки, создаваемой прикладными задачами, а также системными процессами (управление заданиями, задачами, данными, ввод – вывод и др.). Применение статистических методов для этого класса объектов объясняется тем, что хотя рабочая нагрузка, как правило, хорошо наблюдаема, однако по своей природе – это чрезвычайно сложный объект. В нем совмещены свойства прикладных задач, технология обработки данных, организация операционной системы и даже конфигурация ЭВМ, для которой разрабатывается программное обеспечение. Поэтому рабочую нагрузку приходится рассматривать как черный ящик и описывать количественные взаимосвязи статистическими методами. Регрессионные модели применяются также для компактного представления и анализа зависимостей, воспроизводимых на имитационных моделях.

Аналитические методы. Аналитические методы исследования вычислительных систем сводятся к построению математических моделей, которые представляют физические свойства как математические объекты и отношения между ними, выражаемые посредством математических операций. При использовании аналитических методов оператор F , устанавливающий зависимость $Y = F(X)$ между характеристиками и параметрами объекта, представляется совокупностью математических выражений (формул) – алгебраических, дифференциального и интегрального исчисления и др. Модели, построенные этими методами, называются *аналитическими моделями*. В таких моделях зависимость между характеристиками и параметрами может быть представлена в явной аналитической форме – в виде выражений $y_m = f_m(x_1, \dots, x_N)$, решенных относительно искомых величин, или в неявной форме – в виде уравнений $\Phi(Y, X) = 0$, связывающих характеристики и параметры.

При построении аналитических моделей свойства объектов описываются исходя из свойств составляющих – физических элементов или элементарных процессов. Для этого используется подходящий математический аналог и с помощью соответствующего математического аппарата строятся выражения, которые связывают показатели, характеризующие элементы. Последовательным применением математических правил совокупность выражений, моделирующих свойства элементов, сводится к форме, представляющей зависимость $Y = F(X)$ между характеристиками и параметрами всей системы.

Как правило, свойства элементов и систем удается представить в аналитической форме, если принимаются определенные допущения о свойствах и поведении описываемых объектов: независимость одних факторов от других, линейность некоторых зависимостей, мгновенность переходов между состояниями и т. д. Если допущения соответствуют реальности, модель хорошо воспроизводит зависимость между характеристиками и параметрами. Однако во многих случаях допущения приводят к существенным отличиям модели от реального объекта, вследствие чего моделируемая зависимость существенно отличается от реальной и характеристики представляются на модели с большой погрешностью. Так, предположение о том, что процессы обладают марковским свойством, может оказаться ошибочным, что приводит к большим погрешностям марковских моделей и даже к неверным оценкам. Основные аналитические методы теории массового обслуживания базируются на предположении, что интервалы времени между заявками входящих потоков и длительности обслуживания распределены по экспоненциальному закону. Когда это предположение выполняется, аналитические методы позволяют точно оценивать характеристики системы. Если же потоки и длительности существенно отличаются от предполагаемых, моделируемые характеристики могут сколь угодно отличаться от реальных.

Таким образом, аналитические модели, базируясь на допущениях о свойствах объектов, применимы для исследования только тех систем, в отношении которых справедливы принятые допущения. Многие системы из-за специфики своей организации недоступны для исследования аналитическими методами.

Ценность аналитических методов и моделей для теории и практики обусловлена следующими причинами. Во-первых, зависимости, полученные аналитическими методами»,

являются строго доказанными и их достоверность не вызывает сомнений, конечно с учетом принятых при выводе допущений. Поэтому аналитические зависимости используются в качестве своеобразных эталонов, с которыми сопоставляются результаты, получаемые другими методами. Некоторые аналитические зависимости носят абсолютно общий характер и используются в качестве законов, отображающих фундаментальные свойства вычислительных систем. Во-вторых, аналитические модели имеют большую познавательную ценность. Аналитические зависимости определяют характеристики для всей области значений параметров и несут в себе информацию о поведении соответствующих систем при любых сочетаниях параметров. На основе аналитических моделей легко определяются экстремальные и предельные значения характеристик и оцениваются эффекты от изменения параметров. В-третьих, аналитические модели характеризуются наименьшей сложностью вычислений. Это свойство чрезвычайно важно при решении задач синтеза, поскольку оптимизация связана с многократными вычислениями характеристик при различных значениях параметров.

Аналитические методы и модели раскрывают фундаментальные свойства вычислительных систем и составляют ядро теории вычислительных систем.

Имитационные методы. Имитационные методы основаны на представлении порядка функционирования системы в виде алгоритма, который называется *имитационной (алгоритмической) моделью*. Программа содержит процедуры, регистрирующие состояния имитационной модели и обрабатывающие зарегистрированные данные для оценки требуемых характеристик процессов и моделируемой системы.

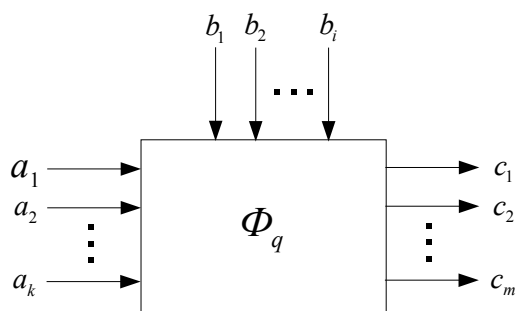


Рис. 7.6. Агрегат как элемент модели

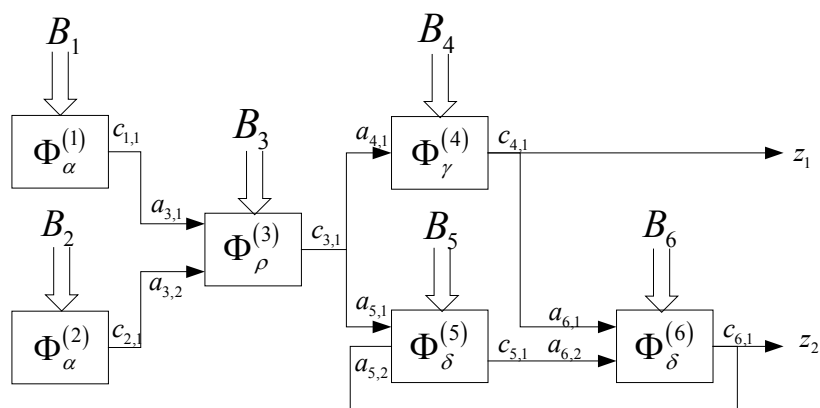


Рис. 7.7. Агрегатная модель

При построении имитационных моделей широко используется агрегатный подход. Для моделирования заданного класса систем создается набор агрегатов Φ_1, \dots, Φ_Q – элементов модели. Агрегаты могут соответствовать элементам систем, например процессорам, оперативным запоминающим устройствам, каналам ввода–вывода, каналам передачи данных и другим, воспроизводя определенные аспекты их функционирования. В

качестве агрегатов могут выступать математические объекты, с помощью которых генерируются и преобразуются необходимые процессы. Так, для моделирования систем на основе сетей массового обслуживания в качестве агрегатов представляются источники потоков заявок, системы массового обслуживания, узлы, управляющие распределением заявок по нескольким направлениям, и т.д. По существу агрегат – описание функции некоторого объекта в аспектах, соответствующих цели моделирования – оценке производительности, надежности и т. д. Функции агрегатов Φ_1, \dots, Φ_Q представляются в параметрической форме, т. е. в записи функций используются параметры, характеризующие конкретный объект. Так, параметром процессора является производительность (быстродействие), оперативной памяти – емкость, системы массового обслуживания – дисциплина обслуживания, число каналов и распределение длительности обслуживания. Функция агрегата $\Phi_q, q = 1, \dots, Q$, представляется в алгоритмической форме – в виде процедуры $\Phi_q = (a_1, \dots, a_k, b_1, \dots, b_l, c_1, \dots, c_m)$, где параметры a_1, \dots, a_k – определяют состояние входов элемента, b_1, \dots, b_l – режим его функционирования и c_1, \dots, c_m – состояние выходов элемента. Сколь ни была бы сложна функция агрегата Φ_q в модели агрегат выглядит как элемент (рис. 7.6), настраиваемый на заданный режим функционирования множеством параметров $B_q = \{b_1, \dots, b_l\}$ и преобразующий входные воздействия $A_q = \{a_1, \dots, a_k\}$ в выходные состояния $C_q = \{c_1, \dots, c_m\}$ в соответствии с функцией агрегата Φ_q и значениями параметров B_q . Множество агрегатов разного типа $\{\Phi_1, \dots, \Phi_Q\}$ составляет базис имитационных моделей заданного класса систем.

Имитационная модель собирается из агрегатов путем соединения выходов агрегатов с входами других агрегатов (рис. 7.7). На рисунке агрегаты обозначены $\Phi_\alpha^{(i)}$, где α – тип и i – порядковый номер агрегата в модели. Агрегаты $\Phi_\alpha^{(1)}$ и $\Phi_\alpha^{(2)}$ – генераторы, формирующие воздействия в соответствии с параметрами B_1 и B_2 . Состав агрегатов, структура связей между ними и наборы параметров агрегатов B_1, \dots, B_6 определяют модель. Процесс моделирования состоит в реализации процедур $\Phi_\delta^{(i)}$ в необходимом порядке. При этом значения, формируемые на выходах агрегатов, переносятся на входы связанных с ними агрегатов, в результате чего вычисляются значения z_1 и z_2 . Путем обработки данных, наблюдаемых в характерных точках модели (на выходах элементов), получают оценки качества функционирования любого из агрегатов и системы в целом.

Имитационные модели воспроизводят процесс функционирования и свойства исследуемых систем исходя из априорно известных свойств элементов системы – за счет объединения моделей элементов в структуру, соответствующую исследуемой системе, и имитации функционирования элементов в их взаимодействии.

Исследование вычислительных систем имитационными методами состоит из нескольких этапов.

1. Определение принципов построения модели. Цель этого этапа – сформировать общий замысел модели (состав характеристик и параметров, подлежащих отображению, область определения модели, требования к точности результатов моделирования, тип математической модели, программные и технические средства для описания и реализации модели). На этом этапе выдвигаются гипотезы о свойствах моделируемой системы, принимаются допущения для использования соответствующих математических методов и конкретизируются эксперименты, проводимые на модели.

2. Разработка модели. Цель этого этапа – создание программы моделирования для ЭВМ. При этом общий замысел модели преобразуется в конкретное алгоритмическое описание. Этап завершается проверкой работоспособности и адекватности модели.

3. Моделирование на ЭВМ. Цель этого этапа – получение с помощью модели данных о поведении исследуемой системы, обработка полученных данных, а при синтезе системы –

выбор параметров, оптимизирующих заданные характеристики системы и удовлетворяющих заданным ограничениям.

Важнейшее свойство метода имитационного моделирования – универсальность, проявляющаяся в следующем. Во-первых, метод имитации позволяет исследовать системы любой степени сложности. Усложнение объекта исследования приводит к увеличению объема данных, вводимых в модель, и времени моделирования на ЭВМ, но при этом принципы построения моделей остаются неизменными. Во-вторых, метод имитации не ограничивает уровень детализации в моделях. С помощью алгоритмов можно воспроизводить любые, сколь угодно своеобразные взаимосвязи между элементами системы и процессы функционирования. Более детальное представление организация и функционирования системы сказывается только на объеме алгоритмического описания модели (программы) и затратах времени на моделирование. Особенности организации и функционирования, препятствующие использованию аналитических методов, легко воспроизводятся в имитационных моделях. В-третьих, имитационная модель является неограниченным источником данных о поведении исследуемой системы – новые эксперименты на модели позволяют получать дополнительные данные о системе. За счет этого гарантируется детальная оценка характеристик, функционирования как системы в целом, так и ее составляющих. Как правило, увеличивая длительность экспериментов на моделях или число экспериментов, т. е. время моделирования, можно добиться высокой точности результатов моделирования.

Недостатки имитационных методов – большие затраты времени на моделирование и частный характер получаемых результатов. В имитационной модели процесс функционирования системы воспроизводится во всех существенных для исследования деталях за счет последовательного выполнения на ЭВМ операций над величинами. Число операций, обеспечивающее воспроизведение представленных интервалов функционирования системы, оказывается значительным и при моделировании систем умеренной сложности составляет 10^8 – 10^{12} операций на одну реализацию модели. Поэтому при моделировании на ЭВМ, имеющей быстродействие миллион операций в секунду, для одного прогона модели требуются минуты и часы процессорного времени. При этом модель позволяет оценить характеристики системы только в одной точке, соответствующей значениям параметров X , введенных в модель перед началом моделирования. Чтобы определить зависимость между характеристиками и параметрами, необходимы многократные прогоны модели, в результате которых значения Y определяются для многих наборов параметров. Возможности методов оптимизации параметров на имитационных моделях ограничиваются большими затратами времени на моделирование системы в одной точке.

Несмотря на указанные недостатки, методы имитационного моделирования в силу их универсальности широко используются при теоретических исследованиях и проектировании вычислительных систем. Имитационные модели позволяют исследователю к разработчику формировать представления о свойствах системы и, познавая систему через ее модель, принимать обоснованные проектные решения.

Экспериментальные методы. Экспериментальные методы основываются на получении данных о функционировании вычислительных систем в реальных или специально созданных условиях с целью оценки качества функционирования и выявления зависимостей, характеризующих свойства систем и их составляющих. Типичные задачи, решаемые экспериментальными методами, – оценка производительности и надежности системы, определение состава и количественных показателей системной нагрузки в зависимости от прикладной нагрузки и т. д.

Экспериментальные исследования выполняются в следующем порядке:

1. Формулируется цель исследования.
2. Выбирается или разрабатывается методика исследования, которая устанавливает модель исследуемого объекта; способ и средства измерения; способ и средства обработки измерительных данных, а также интерпретация результатов измерений и обработок.

3. Проводятся измерения процесса функционирования объекта в реальных или специально создаваемых условиях.

4. Измерительные данные обрабатываются и соответствующим образом интерпретируются.

Экспериментальные методы обеспечивают получение наиболее достоверных данных о вычислительных системах, в чем их преимущество по сравнению с аналитическими и имитационными методами, основанными на использовании моделей. Во многих случаях экспериментальные методы являются единственным источником информации о функционировании и свойствах вычислительных систем. Например, количественная оценка параметров рабочей нагрузки систем общего назначения производится в основном экспериментальными методами. Особенно велико значение экспериментального метода при решении задач эксплуатации, так как совершенствование конфигурации и режима функционирования систем не мыслимо без использования измерительных данных, представляющих конкретные условия работы системы. Недостатки экспериментальных методов – большие затраты труда и времени на проведение экспериментальных исследований, а также частный характер получаемых результатов, распространение которых на системы с другой конфигурацией и режимом функционирования требует достаточно сложной работы.

7.3. ПРИНЦИПЫ АНАЛИЗА ПРОИЗВОДИТЕЛЬНОСТИ

Производительность вычислительных систем общего назначения оценивается в зависимости от области применения номинальной, комплексной, системной производительностью и производительностью на рабочей нагрузке (см. § 1.3). Номинальная производительность характеризует только быстродействие, или производительность устройств, входящих в состав системы. Комплексная производительность учитывает не только быстродействие устройств, но и структуру системы – ее влияние на быстродействие совместно функционирующих устройств. Системная производительность учитывает как вышеназванные факторы – быстродействие устройств и структуру связей между ними, так и влияние операционной системы. Производительность на рабочей нагрузке (кратко – производительность) отображает все факторы, влияющие на системную производительность, и, кроме того, свойства рабочей нагрузки – задач, решаемых вычислительной системой. С производительностью тесно связана такая характеристика качества обслуживания пользователей, как время ответа, т. е. время пребывания задач в системе. Поэтому при оценке производительности определяется не только количество работы, выполняемое системой в единицу времени, но и время ответа для всего множества задач и отдельных классов задач.

Производительность вычислительной системы проявляется, с одной стороны, в скорости обработки задач, а с другой – в степени использования ресурсов системы. Чем больше загружены ресурсы, тем выше производительность системы, и недогрузка ресурсов свидетельствует о наличии резервов для повышения производительности. Поэтому при анализе производительности системы оцениваются не только показатели производительности, но и показатели, характеризующие использование ресурсов.

Производительность вычислительной системы связана с продолжительностью процессов обработки задач, которая зависит от трех факторов: 1) рабочей нагрузки; 2) конфигурации системы; 3) режима обработки задач. Эти три фактора в совокупности определяют порядок развития вычислительных процессов во времени, и первая задача анализа производительности сводится к поиску компактных и информативных форм представления вычислительных процессов. Эти формы создают концептуальную (понятийную) основу для оценки функционирования вычислительных систем в процессе эксплуатации и при исследовании с помощью моделей производительности. Вторая задача анализа – создание моделей, позволяющих прогнозировать производительность систем для различной конфигурации, режимов обработки и, возможно, разной рабочей нагрузки.

Способы описания процессов функционирования. Применительно к задачам анализа производительности функционирование вычислительной системы рассматривается как совокупность процессов, связанных с использованием ресурсов системы. К ресурсам $R = \{R_1, \dots, R_{N+P}\}$ относятся устройства R_1, \dots, R_N , разделяемые между процессами во времени, а также устройства памяти (память) R_{N+1}, \dots, R_{N+P} , разделяемые во времени и по емкости. Процесс характеризуется тройкой параметров: $J = \langle t, A, T \rangle$, где t – момент начала процесса; A – атрибуты, устанавливающие имя источника процесса (пользователя, программы и т.п.), и факторы, влияющие на режим обработки (имя класса, приоритет и др.); T – трасса процесса. Трасса характеризует порядок использования ресурсов и представляется последовательностью событий $T = \{S_1, \dots, S_K\}$, связанных с изменением состояния процесса. Событие S_k характеризуется моментом его возникновения t_k , именем ресурса, с которым связано событие, и параметрами, определяющими использование ресурса (занятие или освобождение, тип операции, выполняемой устройством, емкость выделяемой памяти и т. д.). В трассе фиксируется весь объем данных, отображающих взаимодействие процесса с ресурсами и позволяющих установить порядок обращения к ресурсам и объем их использования. Трасса представляет процесс наиболее полно. Однако у этой характеристики существенный недостаток – большой объем данных. Так, обычно на одну реализацию вычислительного процесса приходится 10^3 – 10^6 обращений к периферийным устройствам, что составляет 10^3 – 10^6 смен состояний процесса или 10^5 – 10^7 байт данных.

$\tau_{ВВ}$	$\omega_{ВХ}$	$\omega_{П}$	$\omega_{Р}$	$\tau_{ПР}$	$\tau_{НМД}$	$\tau_{НМЛ}$	$\omega_{ВЫВ}$	$\tau_{ВЫВ}$
Ввод	Ожидание во входной очереди	Ожидание памяти	Ожидание ресурсов	Процессорная обработка	Работа с НМД	Работа с НМЛ	Ожидание вывода	Вывод

Рис. 7.8. Профиль вычислительного процесса

Более компактная форма представления процесса – *профиль процесса* (рис. 7.8). Для построения профиля выделяются фазы процесса: ввод, ожидание во входной очереди, ожидание памяти и т. д. Реализация процесса представляется в виде последовательности фаз, продолжительность пребывания в которых характеризуется значениями τ_i (время использования устройств) и ω_i (время ожидания). Сумма этих значений составляет время пребывания задания в системе. Профиль процесса дает наглядное представление о продолжительности каждой фазы и соотношении длительности фаз. Для повышения информативности профиль снабжается данными о числе обращений к устройствам, а также о емкости используемой памяти, в том числе о минимальной, средней и максимальной требуемой емкости. Таким образом, профиль определяет время τ_1, \dots, τ_N использования устройств R_1, \dots, R_N и число обращений n_1, \dots, n_N к каждому из устройств, а также потребность в памяти.

Профиль процесса отображает свойства программы и одновременно режима обработки. Свойства программы проявляются в объеме использования ресурсов – времени использования устройств и емкости памяти. Режим обработки сказывается на времени ожидания. В однопрограммном режиме состояние ожидания отсутствует. С увеличением уровня мультипрограммирования время ожидания возрастает, особенно для низкоприоритетных процессов. На время ожидания влияет уровень загрузки системы, который возрастает с интенсивностью поступления заданий на обработку и приводит к увеличению времени пребывания заданий в очередях.

Процессы в вычислительных системах разделяются на два класса: прикладные и системные. *Прикладной процесс* связан с актом обслуживания пользователя и представляет собой некоторую единицу работы – выполнение задания. Прикладные процессы поддерживаются *системными процессами*, порождаемым управляющими программами операционной системы. К системным относятся процессы системного ввода – вывода и супервизорные. Системные процессы, как и прикладные, потребляют ресурсы системы – оперативную и внешнюю память, процессорное время каналы ввода–вывода и НМД. Объем использования ресурса зависит от режима обработки (уровень мультипрограммирования число инициированных процессов системного ввода – вывод и т. д.), а также от параметров заданий (число шагов и описаний наборов данных, число обращений к наборам данных и т. д.). Путем измерений определяется загрузка ресурсов со стороны системных процессов. Потребность системных процессов в ресурсах: обычно выражается в виде уравнений регрессии, аргументами которых являются параметры прикладных процессов.

Способы описания загрузки ресурсов. Производительность системы непосредственно связана с загрузкой устройств. Загрузка устройства – время, в течение которого устройство занято работой, т. е. не простаивает. Если τ_1, \dots, τ_k – длительность рабочих интервалов и T – время работы системы, то загрузка устройств; на отрезке времени T .

$$\rho = \frac{1}{T} \sum_{k=1}^K \tau_k \leq 1 \quad (7.10)$$

Если номинальное быстродействие устройства равно V операций в секунду, то фактическое быстродействие с учетом простоев устройства из-за недогрузки составляет ρV операций в секунду. Если ϑ – среднее число операций, выполняемых устройством при обработке одного задания, то производительность системы, исчисляемая в обрабатываемых за секунду заданиях,

$$\lambda = \rho V / \vartheta \quad (7.11)$$

Следовательно, загрузка устройства характеризует производительность системы с точностью до коэффициента пропорциональности.

При анализе производительности большую роль играет не только значение, но и структура загрузки – составляющие, из которых складывается значение ρ . Типичная структура представлена на рис. 7.9. В данном случае выделено три класса процессов (видов нагрузки): системные процессы, пакетная и оперативная обработка. Два последних относятся к прикладным процессам. Указанные классы процессов создают загрузку ρ_c , ρ_n и ρ_o соответственно.; Эти значения характеризуют время работы устройства, используемое для реализации соответствующих функций. Сумма $\rho = \rho_c + \rho_n + \rho_o$ равна загрузке устройства, а значение $\eta = 1 - \rho$ характеризует простой. Простой η обусловлен двумя факторами. Во-первых, для компенсации пульсаций нагрузки, связанной с оперативной обработкой данных, выделяется резерв, за счет чего обеспечивается приемлемое время ответа при пиковых нагрузках, во-вторых, возможна нехватка других ресурсов, из-за чего рассматриваемое устройство не может быть загружено полностью в рамках данной конфигурации, режима функционирования системы и текущей нагрузки.

Структуру загрузки можно представить более детально, если, например, в составе системных процессов выделить ввод – вывод, пакетную обработку разделить на несколько классов задач и т.д. Структура загрузки всех ресурсов системы дает представление о распределении ресурсов между процессами и о резервах для увеличения производительности и улучшения качества обслуживания пользователей.

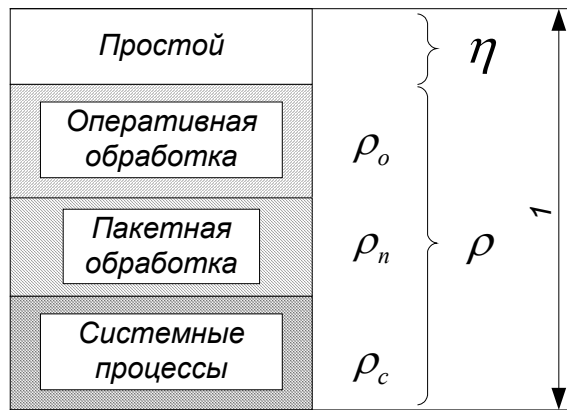


Рис. 7.9. Структура загрузки устройства

Сумма

$$P = \sum_{n=1}^N \rho_n \quad (7.12)$$

называется загрузкой вычислительной системы.

Если система функционирует в однопрограммном режиме, причем не простаивает из-за отсутствия нагрузки, $P = 1$. Если $P > 1$, то производительность системы в этом режиме в P раз выше, чем в однопрограммном. Таким образом, загрузка системы характеризует производительность системы по отношению к производительности однопрограммного режима.

Эффективный способ повышения производительности вычислительных систем – мультипрограммирование, позволяющее совместить во времени работу многих устройств, в результате чего увеличивается загрузка каждого устройства, а следовательно, загрузка и производительность системы. Для анализа степени совмещения, а также для выбора конфигурации и режима обработки используется *профиль загрузки системы* (диаграммы Ганта).

Профиль загрузки системы строится следующим образом. Пусть S_n – состояние устройства: $S_n = 0$, если устройство не загружено (простаивает), и $S_n = 1$, если устройство занято работой. Состояние системы будем характеризовать вектором $S = (S_1, \dots, S_N)$, составляющие которого – состояния устройств R_1, \dots, R_N . Система, содержащая N устройств, может пребывать в 2^N состояниях $(0, \dots, 0, 0)$, $(0, \dots, 0, 1)$, $(0, \dots, 1, 0)$, ..., $(1, \dots, 1)$. Первое из них соответствует простоя и последнее – одновременной работе всех устройств. Состояние системы будем нумеровать числами, представляющими десятичные значения двоичных наборов (S_1, \dots, S_N) : $S_0 = (0, \dots, 0, 0)$, $S_1 = (0, \dots, 0, 1)$, ..., $S_M = (1, \dots, 1, 1)$, где $M = 2^N - 1$.

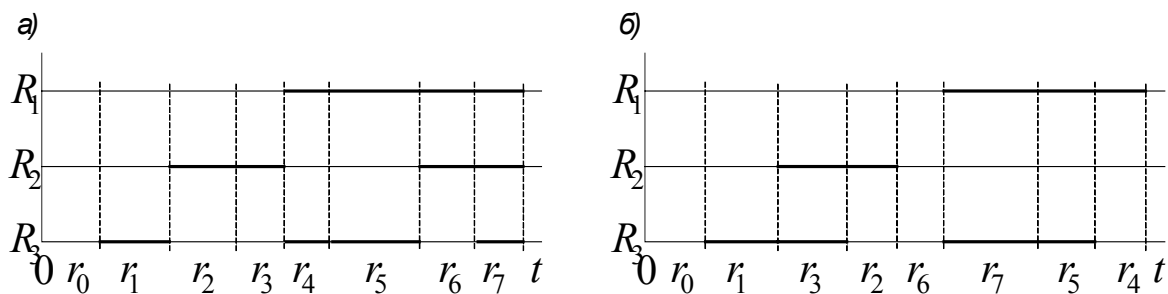


Рис. 7.10. Профиль загрузки системы

Пусть за время T система находилась в состояниях (S_0, \dots, S_M) на протяжении интервалов (τ_0, \dots, τ_M) соответственно. Значение $r_m = \tau_m / T, m = 0, \dots, M$, характеризует долю времени, в течение которого система находилась в состоянии S_m , а также вероятность этого состояния. Распределение вероятностей состояний системы наиболее наглядно представляется в графической форме, на пример для трех устройств (рис. 7,10,с). Каждому устройству R_1, R_2, R_3 соответствует ось длиной 1. Оси разделяются на отрезки длиной r_0, \dots, r_7 , соответствующие интервалам времени, на протяжении которых система пребывает в состояниях S_0, \dots, S_7 . Отрезки r_m выделяются жирной линией, если в состоянии S_m устройство $R_n, n=1, 2, 3$, загружено работой. Для устройства R_1 такими являются отрезки r_4, \dots, r_7 , а для устройства R_2 – отрезки r_1, r_3, r_5 и r_7 . Полученная таким способом диаграмма называется профилем загрузки системы. Из профиля загрузки видна степень совмещения работы разных устройств во времени. В данном случае очевидно, что возможна совместная работа устройств R_1 и R_2 и одиночная работа устройства R_2 , а работа устройства R_1 в основном совмещается с работой остальных устройств. Суммарная длина отрезков, выделенных на осях R_1, R_2, R_3 , характеризует соответственно загрузку ρ_1, ρ_2, ρ_3 устройств.

Наглядность профиля загрузки возрастает, если состояния S_m и соответствующие им отрезки r_m размещать в порядке, определяемом кодом Грея. При этом уменьшается число разрывов между отрезками, выделенными жирными линиями. Так, трехразрядный код Грея порождает последовательность кодов 000, 001, 011, 010, 110, 111, 101, 100, которой соответствует последовательность состояний $r_0, r_3, r_2, r_6, r_7, r_5, r_4$. Профиль загрузки системы, построенный на основе кода Грея и эквивалентный рассмотренному выше, изображен на рис. 7.10, б. Данный профиль содержит только один разрыв между отрезками занятости, в то время как предыдущий – четыре разрыва.

Профиль загрузки системы, отображающий каждое из 2^N состояний, где N – число устройств, называется *полным*. Сложность полного профиля растет как показательная функция числа устройств N . При $N \geq 5$ число отрезков r_0, \dots, r_M оказывается большим и профили теряют наглядность. Поэтому для отображения загрузки устройств часто используются *неполные профили*, в которых представляются лишь наиболее существенные состояния.

Модели производительности. При проектировании, а также при совершенствовании конфигурации и режимов функционирования систем, находящихся в эксплуатации, возникает необходимость оценивать производительность различных вариантов. Для этого используются модели производительности вычислительных систем, позволяющие к тому же оценивать характеристики процессов и использования ресурсов. Характеристики необходимы для выявления факторов, влияющих на производительность, а также узких мест и недоиспользованных ресурсов, т. е. в конечном итоге – для выбора подходящего варианта.

Имитационные модели производительности систем общего назначения состоят из трех основных блоков (рис. 7.11): рабочей нагрузки, планирования работ и выполнения задач. Модель рабочей нагрузки создает потоки заданий, формируемых пользователями на входе системы, и определяет параметры заданий. Модель настраивается на конкретный тип нагрузки набором параметров B_1 . Модель планирования работ воспроизводит обеспечение заданий ресурсами. Модель настраивается на конкретный режим обработки набором параметров B_2 (число разделов или инициаторов, распределение классов задач между инициаторами и т. д.). Задания, обеспеченные на фазе планирования ресурсами, образуют задачи, обработка которых воспроизводится моделью выполнения задач. Набор параметров B_3 характеризует структуру системы и быстродействие устройств, влияющие на продолжительность выполнения задач. Состояние процессов в общем случае влияет на состояние процессов планирования и порядок поступления задач в систему: данные о состоянии последующих фаз обработки передаются в предыдущие фазы (штриховая линия на рисунке).

Для разных целей необходимы различные модели производительности, отличающиеся составом воспроизводимых параметров и точностью воспроизведения характеристик.

Наиболее сложны модели, которые предназначены для выбора режима обработки, согласованного с рабочей нагрузкой и обеспечивающего заданное качество обслуживания – максимальную пропускную способность при заданных ограничениях на время ответа. В таких моделях приходится детально воспроизводить конфигурацию системы, состав рабочей нагрузки и параметры операционной системы, посредством которых она настраивается на заданный режим обработки. Точность оценки характеристик должна быть высокой. Повышаются требования к информативности: из модели должны выводиться подробные данные о профиле процессов и загрузке ресурсов, в том числе о структуре загрузки.

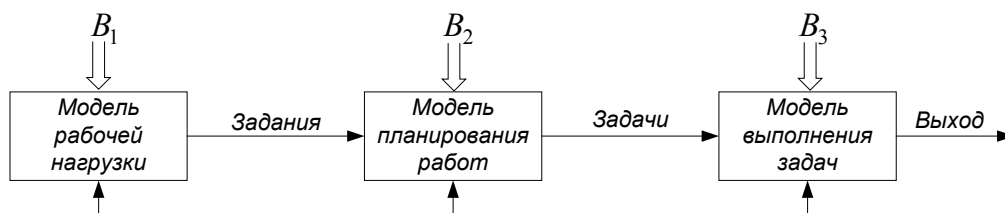


Рис. 7.11. Состав модели производительности

Для выбора конфигурации при совершенствовании эксплуатируемых систем и проектировании новых используются более простые модели, воспроизводящие рабочую нагрузку и режим работы лишь в общих чертах. Основное внимание уделяется моделированию фазы выполнения задач, влияние которой на производительность наиболее велико. В этом случае оценка производительности с погрешностью 20–25 % вполне приемлема. Более того, при выборе конфигурации модель считается адекватной системе, если воспроизводит хотя бы тенденцию изменения производительности, т. е. прирост или снижение ее при изменении конфигурации системы.

Состав процессов, воспроизводимых моделью, и точность их воспроизведения существенно зависит от априорных данных о моделируемой системе, которыми располагает исследователь. Модель рабочей нагрузки не может воспроизвести свойства нагрузки, о которых мы не имеем четкого представления. То же самое можно сказать и о воспроизводимости процессов управления обработкой и процессов выполнения задач. Состав данных о рабочей нагрузке и функционировании системы предопределяет воспроизводимые в модели закономерности и точность оценок. Недостаточность данных о моделируемой системе является более существенным препятствием при построении моделей, чем ограниченные возможности методов моделирования и математических моделей.

Модели производительности могут строиться как имитационные, аналитические или статистические. Из-за существенного различия этих классов моделей по точности и затратам на реализацию могут использоваться различные методы при построении элементов модели производительности: рабочей нагрузки, фаз планирования работ и выполнения задач. Такие модели производительности называются *гибридными*. В типичной гибридной модели рабочая нагрузка и планирование воспроизводятся имитационными, а выполнение задач – аналитическими методами.

7.4. МЕТОДЫ И СРЕДСТВА ИЗМЕРЕНИЙ И ОЦЕНКИ ФУНКЦИОНИРОВАНИЯ

Измерения являются источником наиболее достоверных данных о функционировании вычислительных систем и проводятся в следующих целях: 1) для учета выполненных работ; 2) для оценки функционирования; 3) для идентификации вычислительной системы – построения моделей. Измерения могут быть направлены на исследование как системы в целом, так и отдельных подсистем.

Схема измерений представлена на рис. 7.12. Объектом измерений является вычислительная система, функционирующая, как правило, в рабочем режиме. К системе подключаются измерительные средства – мониторы, реагирующие на изменение состояний системы и измеряющие параметры состояний (моменты изменения состояний, продолжительность пребывания в них и др.). Измерительные данные поступают от мониторов в архив на протяжении заданного промежутка времени, накапливаются и затем обрабатываются.

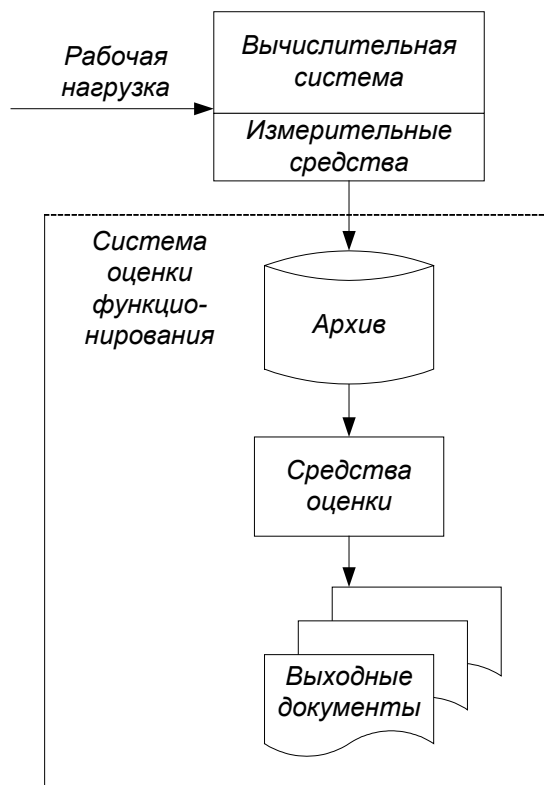


Рис. 7.12. Организация измерений и оценки функционирования

Функционирование вычислительной системы проявляется в изменении состояний процессов и ресурсов. Состояния отображаются в управляющих таблицах, которые формируются управляющими программами операционной системы, и в первую очередь супервизором. Состояния процессов и ресурсов изменяются в моменты выполнения специальных команд – обращения к супервизору и привилегированных, с помощью которых супервизор управляет процессами и ресурсами, а также при поступлении сигналов прерывания, извещающих супервизор о моментах окончания операций ввода – вывода, особых ситуациях в системе и сигналах на ее входах. Наряду с этим состояния устройств отображаются соответствующими электрическими сигналами.

Мониторы строятся с использованием различных методов измерений и средств и классифицируются в зависимости от этого (рис 7.13).

Трассировочный и выборочный методы измерений. *Трассировочный метод* измерений основан на регистрации событий, соответствующих моментам изменения состояний вычислительной системы. К таким событиям, в частности, относятся начало и конец ввода задания, шага задания, этапа процессорной обработки, обращения к внешней памяти и т. д. События регистрируются монитором в виде событийного набора данных T (рис. 7.14), состоящего из последовательности записей s_1, s_2, \dots , соответствующих последовательности событий. В записи регистрируется момент возникновения события, имена процесса и ресурса, с которыми оно связано, и параметры события, – например емкость занимаемого или освобождаемого блока памяти, число передаваемых байтов данных и т. д. Событийный набор данных, создаваемый монитором, содержит информацию о

процессах J_1, J_2, \dots и одновременно о ресурсах. На рисунке изображена диаграмма использования устройства R_i , представляющая его состояния (0 – свободно и 1 – занято), и диаграмма использования памяти M_j , характеризующая суммарную емкость, занятую процессами. Мониторы, измеряющие процесс функционирования системы трассировочным методом, называются *трассировочными*.

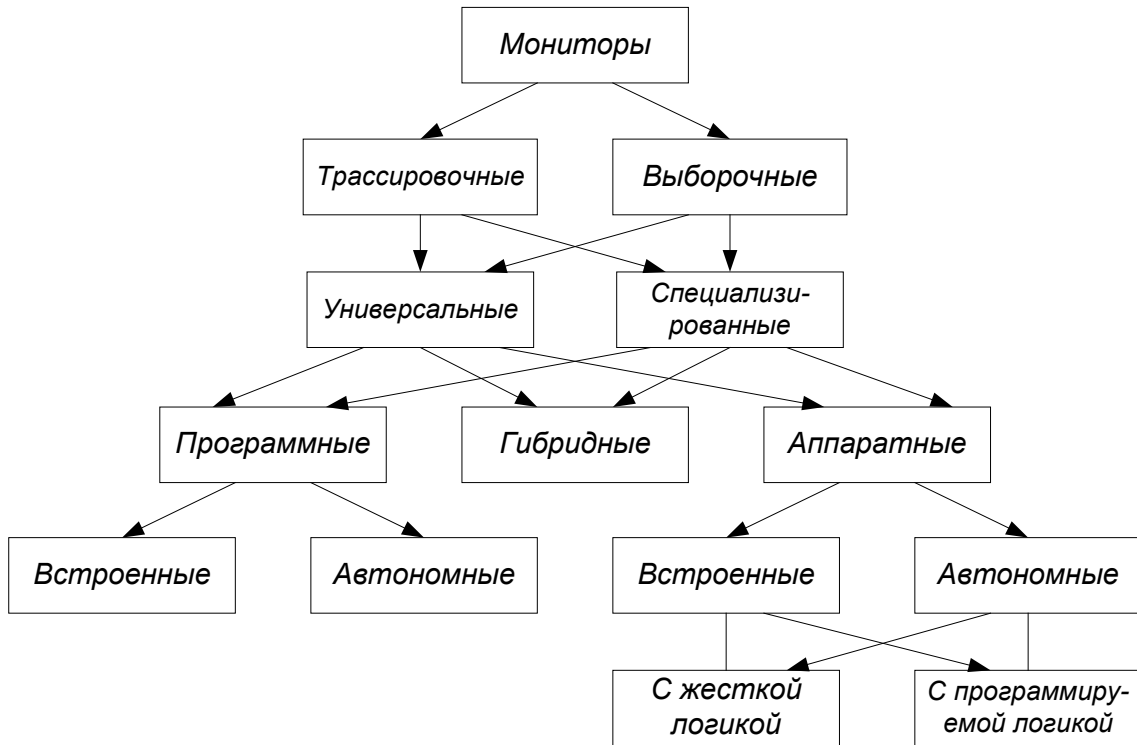


Рис. 7.13. Классификация мониторов

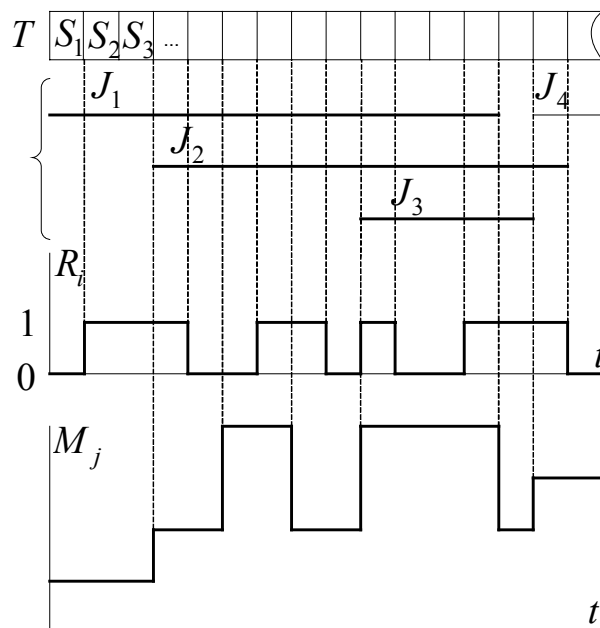


Рис. 7.14. Состав измерительных данных

Выборочный метод измерений основан на регистрации состояний вычислительной системы в заданные моменты времени, как правило, через промежутки длительностью δ . В моменты $t = n\delta, n = 0, 1, 2, \dots$, выборочный монитор регистрирует состояние системы, фиксируя в соответствующих записях данные из управляющих таблиц, или значения электрических сигналов, характеризующих состояния устройств системы. Полученные данные позволяют с погрешностью не более 26 оценивать продолжительность пребывания процессов и ресурсов в различных состояниях и вероятности состояний. Последние определяются значениями $p_i = n_i/n$, где n_i – число выборок, при которых было зарегистрировано состояние i , и n – длительность процесса измерений, определяемая числом выборок.

Трассировочные мониторы измеряют отдельные процессы, например обработку одного задания, более точно, чем выборочные. Однако, если функционирование системы оценивается статистическими методами, выборочный монитор обеспечивает такую же точность, как и трассировочный, правда при большей продолжительности измерений. Основное достоинство выборочных мониторов – возможность измерений сколь угодно быстрых процессов при ограниченном быстродействии.

Универсальные и специализированные мониторы. В зависимости от регистрируемого состава событий (состояний) мониторы подразделяются на универсальные и специализированные.

Универсальный монитор регистрирует все события (состояния) или подавляющее большинство их, благодаря чему событийный набор данных достаточен для построения трасс процессов и использования ресурсов. Объем измерительных данных чрезвычайно велик и составляет 10^5 – 10^7 байт на один процесс. Поэтому в мониторы встраиваются средства настройки, позволяющие регистрировать часть событий, соответствующих целым исследованиям. Универсальные мониторы используются в основном периодически для оценки, например, конкретных системных или прикладных процессов.

Специализированный монитор регистрирует определенную часть событий (состояний), соответствующих конкретной цели измерений, что приводит к умеренному объему измерительных данных и снижает сложность их обработки. Специализированные мониторы широко применяются для учета выполненных работ и оценки загрузки ресурсов. Благодаря умеренному потоку данных специализированные мониторы используются как постоянно действующие измерительные средства для оценки функционирования систем в течение всего рабочего периода.

Программные мониторы. Мониторы, реализованные в виде программы, выполняемой вычислительной системой, называются *программными*.

Программные мониторы трассировочного типа реагируют на определенный класс событий за счет «перехвата» обращений к супервизору, сигналов пребывания и других сигналов, что приводит к передаче управления программным блокам монитора. Приняв управление, монитор выбирает из управляющих таблиц супервизора, программ управления заданиями и данными необходимую информацию, формирует запись, соответствующую событию, и помещает ее в заданную область памяти. Затем передает управление в точку, перед обращением к которой включается в работу в момент возникновения событий, соответствующих смене состояний.

Программные мониторы выборочного типа включаются в работу по таймеру, отмечающему интервалы времени δ , через которые производится опрос состояний вычислительной системы. Как и мониторы трассировочного типа, они получают необходимые данные о состоянии системы из управляющих таблиц и заносят сформированные данные в заданную область памяти.

В зависимости от местоположения и от статуса программные мониторы подразделяются на встроенные и автономные (см. рис. 7.13).

Встроенный программный монитор – совокупность программных блоков, входящих в состав управляющих программ операционной системы. Встроенный монитор создается

совместно с операционной системой и является ее частью. За счет этого обращение к блокам монитора реализуется короткими цепочками команд и минимизируются затраты процессорного времени на выполнение измерительных процедур. Встроенные программные мониторы, как правило, имеют статус управляющих программ операционной системы. В операционные системы встраиваются специализированные измерительные средства для учета выполненных работ, контроля использования ресурсов и получения данных о сбоях и отказах системы. Такого рода измерительные средства дают минимальные сведения о функционировании вычислительной системы.

Для расширения измерительных возможностей используются *автономные программные мониторы* – измерительные программы, выполняемые системой в основном как прикладной процесс. Автономные мониторы загружаются в оперативную память как прикладные программы. Монитор программно связывается с супервизором, за счет чего в момент возникновения событий монитору передается управление. Монитор выбирает необходимые данные из управляющих таблиц, обрабатывает их, формирует запись в наборе измерительных данных и возвращает управление супервизору. Автономные мониторы, как правило, универсальны и позволяют регистрировать широкую номенклатуру событий при трассировочном и состояний и состояний при выборочном методе измерений. Специализированные автономные мониторы используются для контроля за функционированием отдельных подсистем вычислительной системы, например процессов ввода – вывода, работы внешних запоминающих устройств и др. В отличие от встроенных автономные мониторы используются для оценки функционирования системы лишь периодически.

Основное достоинство программного способа построения мониторов – возможность получения сколь угодно детальной информации. Недостатки – зависимость программных мониторов от типа ЭВМ и операционной системы, а также влияние монитора на временные аспекты функционирования системы. Программные мониторы создаются для ЭВМ и операционных систем конкретных типов. Поэтому постановка на ЭВМ новой операционной системы или расширение операционной системы требует модификации измерительных средств. Поскольку программы монитора реализуются совместно с прикладными и системными программами, обработка заданий при измерениях растягивается во времени. Встроенные программные мониторы характеризуются незначительной ресурсоемкостью и порождаемая ими нагрузка на процессор невелика. Автономные программные мониторы имеют значительную ресурсоемкость, которая может составлять 10-15% процессорного времени. Выборочные мониторы позволяют снижать нагрузку на ресурсы за счет увеличения периода регистрации, однако при этом для получения представительного объема данных приходится увеличивать продолжительность измерений.

Аппаратные мониторы. *Аппаратный монитор* – комплекс технических и, возможно, программных средств, предназначенных для измерения процессов функционирования вычислительных систем. Принцип измерений с помощью аппаратного монитора иллюстрируется рис. 7.15. Монитор получает информацию о состоянии системы посредством электрических сигналов, характеризующих состояние отдельных устройств и блоков. Для измерений выявляются точки подключения монитора к системе, в которых присутствуют сигналы, представляющие состояние устройств. В качестве точек подключения наиболее часто используются выходы триггеров и линии интерфейсов. Точки подключения связываются с монитором при помощи зондов. Зонд состоит из усилителя, обеспечивающего передачу сигнала по длинной линии и имеющего высокое входное сопротивление, и линии, соединяющей усилитель с входом монитора. Сигналы с зондов обрабатываются селектором – схемой, формирующей на основе входных сигналов сигналы состояний (событий) s_1, \dots, s_M , которые должны обрабатываться монитором.

Сигналы s_1, \dots, s_M с селектора поступают в измерительный блок, в котором выполняются типичные измерительные процедуры: определяется время поступления сигнала, длительность промежутка между двумя событиями, отмечаемыми соответствующими сигналами, и число событий. Результаты измерений вводятся в микро-ЭВМ по сигналам прерывания или по таймеру. Микро-ЭВМ

обрабатывает поступающие измерительные данные, записывает их на некоторый носитель (например, на магнитную ленту) и оперативные оценки функционирования отображает на терминале. Оперативные оценки используются для контроля за ходом процесса изменений и функционированием вычислительной системы. Зарегистрированные на носителе измерительные данные в дальнейшем подвергаются обработке с помощью ЭВМ монитора или другой ЭВМ, в том числе исследуемой.

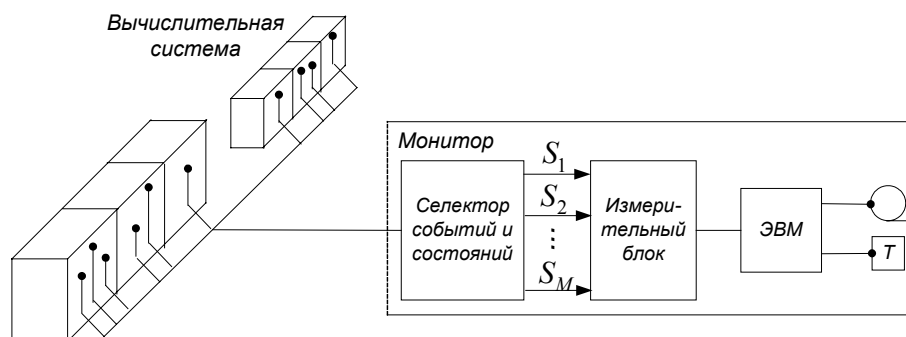


Рис. 7.15. Измерения с помощью аппаратного монитора

В зависимости от назначения и способа построения аппаратные мониторы подразделяются на автономные, встроенные, с жесткой и программируемой логикой (см. рис. 7.13). *Автономный аппаратный монитор* – переносное устройство для измерений различных вычислительных систем. *Встроенный аппаратный монитор* – устройство или блок, конструктивно встроенный в вычислительную систему. *Аппаратные мониторы с жесткой логикой* предназначены для получения фиксированного набора оценок функционирования, который вычисляется с помощью схемных средств или программ, хранимых в постоянной памяти. Мониторы с жесткой логикой наиболее широко используются для контроля за использованием ресурсов системы, оцениваемым с помощью коэффициентов загрузки. *Аппаратные мониторы с программируемой логикой* имеют в своем составе ЭВМ, путем программирования которой можно получать различные данные о функционировании исследуемой системы. В таких мониторах возможна перестройка функций селектора событий и состояний и функций измерительного блока в зависимости от состава входных сигналов и цели измерений.

Пропускная способность аппаратного монитора, определяемая затратами времени на измерение и регистрацию состояний, ограничивается быстродействием используемых в мониторе средств обработки и хранения измерительных данных. Если частота изменения состояний в системе не превосходит пропускной способности монитора, используется трассировочный метод измерений. В противном случае измерения проводятся по выборочному методу, который не налагает ограничений на скорость процессов в измеряемой системе.

Основные достоинства аппаратных мониторов – системная независимость и независимость процесса функционирования системы от процедуры измерений. Системная независимость обусловлена тем, что аппаратные мониторы прямо не связаны с программными средствами, а поэтому могут использоваться для измерений функционирования любых ЭВМ, работающих в различных режимах под управлением любых операционных систем. Аппаратные мониторы не используют ресурсов исследуемой системы, и поэтому процесс измерений не влияет на функционирование системы. Недостаток аппаратного способа измерений – существенные ограничения на состав информации о функционировании системы, доступный для монитора. Монитор может получать только ту информацию, которая отображается в устройствах в виде сигналов, и ему недоступна информация, формируемая программами и отображаемая в памяти системы. Поэтому аппаратные мониторы не могут регистрировать атрибуты заданий (имена пользователей и программ), состояние очередей и другую информацию. Наиболее доступна для регистрации информация, связанная с использованием ресурсов: загрузка устройств, интенсивность обращения к устройствам, частота различных операций, интенсивность потоков данных, передаваемых через интерфейсы, и др.

Гибридные мониторы. Для использования преимуществ программного и аппаратного способа измерений создаются гибридные мониторы, в которых используются программные средства

для получения данных о состояниях системы и аппаратные средства для регистрации измерительных данных, поступающих от программных средств.

В структурном отношении гибридный монитор состоит из программной и аппаратной части. Программная часть – совокупность программных блоков, фиксирующих изменение состояний прикладных и системных процессов. Программные блоки формируют данные для аппаратной части монитора, которые выводятся через соответствующий канал (интерфейс) ввода – вывода. Аппаратный монитор подключается к каналу и, получая данные от измерительных программ, обрабатывает их собственными средствами. За счет такой организации измерительных средств обеспечивается доступ к информации, формируемой на программном уровне, и существенно снижаются затраты ресурсов системы на измерения, поскольку измерительные программы занимают небольшую область памяти и выполняются с незначительными затратами процессорного времени. Аппаратный монитор имеет статус периферийного устройства и работает в основном автономно, используя собственную память, процессор и средства ввода – вывода.

Организация оценки функционирования. Оценка функционирования вычислительных систем сводится к обработке измерительных данных, зарегистрированных программными и аппаратными мониторами, с целью определения системных характеристик (производительность, время ответа и надежность), показателей использования ресурсов, характеристик рабочей нагрузки, а также с целью идентификации системы. Наиболее широко используется двухэтапный способ оценки функционирования (см. рис. 7.12). На первом этапе собираются и накапливаются в архиве измерительные данные. На втором этапе данные обрабатываются. Программные средства хранения, доступа к данным и оценки функционирования совместно с положенными в их основу концептуальными и математическими моделями функционирования и методами измерений образуют *систему оценки функционирования* (рис. 7.16).

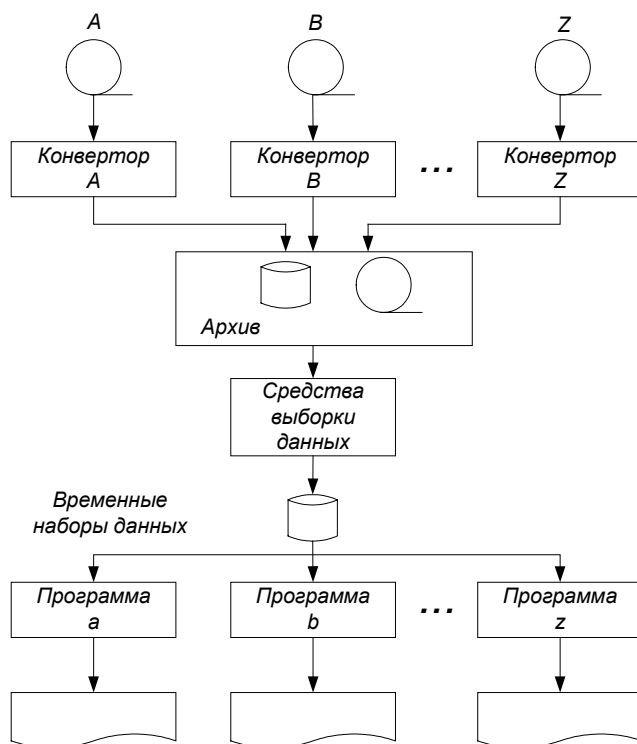


Рис. 7.16. Состав системы оценки функционирования

Средства хранения создают наборы измерительных данных. Как правило, измерительные данные, формируемые монитором, представляют собой событийные наборы, в которых записи соответствуют отдельным событиям. С целью экономии памяти, используемой для хранения данных, событийные наборы перед записью в архив преобразуются в наборы с объектной структурой, в которых записи соответствуют таким

объектам, как задание. Объектные наборы, как и событийные, имеют последовательную организацию с упорядоченными во времени записями об объектах. Преобразование событийных наборов в объектные выполняется программами конвертирования, каждая из которых соответствует монитору определенного типа.

Оценка функционирования системы в большинстве случаев производится на основе некоторого подмножества хранимых в архиве измерительных данных, относящихся к определенным периодам работы системы или к определенным классам заданий. Селективная выборка данных из архива выполняется программными средствами выборки, которые затем формируют наборы данных об определенных классах объектов (временные интервалы, группы пользователей, продолжительность пребывания заданий в системе и др.). Сформированные наборы данных обрабатываются программами оценки системных характеристик, загрузки ресурсов, рабочей нагрузки и т. д. Результаты оценки представляются в виде выходных документов. Для оценки характеристик используются как специально разрабатываемые программы, так и пакеты программ общего применения, например программ статистического анализа.

7.5. МОДЕЛИ РАБОЧЕЙ И СИСТЕМНОЙ НАГРУЗКИ

Эффективность вычислительной системы существенно зависит от степени согласованности конфигурации и режима функционирования системы с нагрузкой, создаваемой прикладными задачами и операционной системой. Поэтому для решения задач системной и технической эксплуатации и при проектировании систем необходимо располагать сведениями о потребностях задач в ресурсах системы. Для получения этих сведений существует два способа:

- 1) анализ программ, в процессе которого оценивается потребность в памяти, процессорном времени и других ресурсах в расчете на одну реализацию каждой программы;
- 2) измерения процесса функционирования ЭВМ и обработки измерительных данных.

Анализ программ с целью определения создаваемой ими нагрузки - достаточно сложный процесс, требующий детальных сведений, во-первых, о структуре программы и, во-вторых, о распределении значений исходных величин, влияющих на порядок выполнения программы. Поэтому оценка нагрузки путем анализа программ производится только для специализированных систем, реализующих небольшое число относительно несложных программ.

Оценка нагрузки на основе измерительных данных – наиболее широко используемый способ анализа нагрузки систем общего назначения и проблемно-ориентированных. При этом проводятся измерения процесса функционирования ЭВМ в течение представительного промежутка времени, охватывающего выполнение 1– 2 тыс. заданий. Полученные данные обрабатываются применительно к выбранной модели нагрузки, в результате чего оцениваются параметры нагрузки, воспроизводимые моделью.

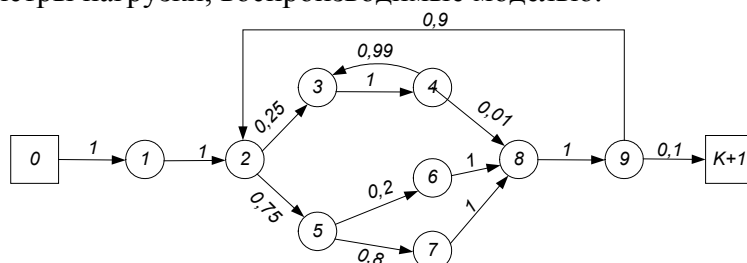


Рис. 7.17. Граф программы

Марковская модель программы. Для определения нагрузки, создаваемой программой в отношении устройств системы, используется марковская модель программы. Как правило, модель представляет собой граф, в вершинах которого, соответствующих операторам программы, отмечены объемы ресурсов, используемых при выполнении

оператора, а на дугах – вероятностных переходов к следующим операторам. Пример графа программы приведен на рис. 7.17. Вершины графа обозначены номерами 0, 1, ..., K, K+1... Вершина 0 – начальная, а вершина K+1 – конечная. Если R_1, \dots, R_N – устройства вычислительной системы, то каждой из вершин $k=1, \dots, K$ ставится в соответствие вектор (g_{1k}, \dots, g_{Nk}) , определяющий потребность оператора k в ресурсах устройств. Значения g_{1k}, \dots, g_{Nk} могут задаваться в виде объемных характеристик ресурса (число процессорных операций, вводимых и выводимых символов или записей, обращений к внешним запоминающим устройствам и т. д.) или в виде временных характеристик (время использования процессора, устройств ввода – вывода и других устройств системы). Если из оператора выходит единственная дуга, то переход по ней происходит с вероятностью i . Если из оператора k выходит несколько дуг $(k, l), (k, m), \dots, (k, \omega)$, то выбор направления перехода рассматривается как случайное событие, характеризуемое вероятностями исходов $p_{ki}, p_{km}, \dots, p_{k\omega}$, причем $p_{ki} + p_{km} + \dots + p_{k\omega} = 1$. Вероятности определяются путем анализа операторов переходов и циклов, влияющих на пути вычислительного процесса. Так, если вероятность выполнения условия в операторе перехода равна 0,25, то двум путям развития вычислительного процесса соответствуют вероятности 0,25 и 0,75. Если цикл повторяется в среднем 100 раз, то вероятность выхода из цикла равна 0,01, а возврата в начало цикла – 0,99.

На основе графа программы строится поглощающая марковская цепь, определяющая порядок выполнения программы. При этом операторам 1, ..., K программы ставятся в соответствие невозвратные состояния s_1, \dots, s_K марковской цепи, а конечной вершине графа (K+1) – поглощающее состояние s_0 . Расчет характеристик поглощающей марковской цепи определяется среднее число попаданий процесса $(\bar{n}_1, \dots, \bar{n}_K)$ в состояния s_1, \dots, s_K и дисперсия числа попаданий $\sigma_1^2, \dots, \sigma_K^2$. На основе полученных значений и заданных (g_{1k}, \dots, g_{Nk}) потребностей операторов в ресурсах определяются средние значения потребности программы в ресурсах:

$$\Theta_n = \sum_{k=1}^K g_{nk} \bar{n}_k, \quad n = 1, \dots, N$$

Значения $\Theta_1, \dots, \Theta_N$ характеризуют использование ресурсов в объемных или временных единицах при одной реализации программы. Если потребности операторов в ресурсах определены во временных единицах, то значение $\Theta = \sum \Theta_n$ характеризует среднее время выполнения программы.

Марковская модель хорошо воспроизводит ресурсные свойства программ. Погрешности оценок связаны в первую очередь с ошибками в определении вероятностей переходов p_{ij} , обусловленными приближенностью априорных сведений о распределении значений исходных данных.

Оценка рабочей нагрузки по измерительным данным. Рабочая нагрузка вычислительных систем общего назначения оценивается на основе измерений процесса функционирования. Для оценки рабочей нагрузки выбирается учетная единица работ: при пакетной обработке – задание, а при оперативной – взаимодействие пользователя с системой, называемое *транзакцией*. На основе измерительных данных, полученных с помощью монитора, формируется объектный набор данных, каждая запись которого характеризует одну учетную единицу работы, например отдельное задание. В зависимости от типа монитора состав данных в записи может быть различным. Так, запись может содержать трассу процесса выполнения работы либо менее детальные сведения, соответствующие некоторой модели процесса. Эти данные используются для оценки рабочей нагрузки – определения ее состава и параметров.

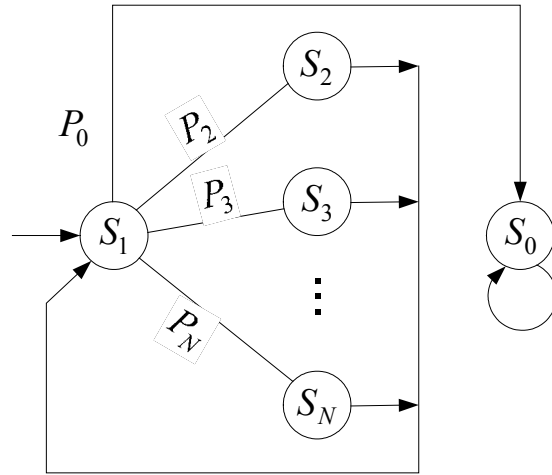


Рис. 7.18. Модель центрального обслуживания

Для повышения достоверности оценки необходима обширная статистика – объемом 1–2 тыс. заданий. Чтобы описать рабочую нагрузку в компактной форме, следует, во-первых, процесс выполнения каждого задания характеризовать небольшим числом параметров и, во-вторых, множество заданий представлять ограниченным числом классов заданий, характеризуя свойства каждого класса среднестатистическим заданием. Для компактного описания процессов выполнения заданий наиболее широко используется модель центрального обслуживания.

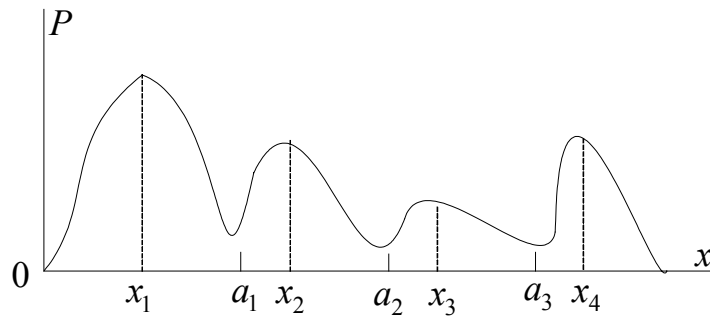


Рис. 7.19. Распределение параметра заданий для неоднородной нагрузки

Модель центрального обслуживания. В модели центрального обслуживания процесс выполнения программы представляется поглощающей марковской цепью с множеством состояний s_0, s_1, \dots, s_N , где s_0 – поглощающее состояние, а s_1, \dots, s_N – невозвратные состояния, соответствующие этапам выполнения процесса на устройствах R_1, \dots, R_N (процессор и периферийные), причем состояние s_1 отождествляется с этапом процессорной обработки.

Матрица вероятностей переходов для марковской цепи

$$P = \begin{matrix} & \begin{matrix} s_0 & s_1 & s_2 & \dots & s_N \end{matrix} \\ \begin{matrix} s_0 \\ s_1 \\ s_2 \\ \dots \\ s_N \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ p_0 & 0 & p_2 & \dots & p_N \\ 0 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 1 & 0 & \dots & 0 \end{bmatrix} \end{matrix} \quad (7.13)$$

и распределение вероятностей начальных состояний $\pi_0 = (0, 1, 0, \dots, 0)$. Граф марковской цепи представлен на рис. 7.18. Процесс начинается этапом процессорной обработки, которому соответствует состояние S_1 . По окончании этого этапа с вероятностью p_2, \dots, p_N процесс переходит в состояние s_2, \dots, s_N , соответствующие работе периферийных устройств R_2, \dots, R_N , т. е. вводу – выводу. По окончании этапа ввода–вывода следует очередной этап процессорной обработки, затем переход к очередному этапу ввода – вывода. Процесс продолжается до тех пор, пока с вероятностью p_0 не перейдет в поглощающее состояние, в котором и заканчивается.

Параметры p_0, p_2, \dots, p_N цепи (7.13) рассчитываются по среднему числу обращений $\bar{n}_2^*, \dots, \bar{n}_N^*$ к периферийным устройствам R_2, \dots, R_N (при одной реализации процесса) следующим образом. Из рис. 7.13 видно, что среднее число этапов процессорной обработки

$$\bar{n}_1^* = \sum_{i=2}^N \bar{n}_i^* + 1$$

С учетом этого

$$p_0 = 1/\bar{n}_1^*; \quad p_i = \bar{n}_i^* / \bar{n}_1^*; \quad i = 2, \dots, N \quad (7.14)$$

Продолжительность пребывания процесса в состояниях s_1, \dots, s_N характеризуется распределениями длительности этапов процессорной обработки $p(\tau_1)$ и этапов ввода–вывода $p(\tau_2), \dots, p(\tau_N)$ или средней длительностью $\bar{\tau}_1, \dots, \bar{\tau}_N$ и средними квадратическими отклонениями $\sigma_1, \dots, \sigma_N$, оцениваемыми по результатам измерений. Когда на основе измерительных данных получены средние значения времени работы $\mathcal{G}_1, \dots, \mathcal{G}_N$ на устройствах p_1, \dots, p_N , средняя длительность этапов процессорной обработки и ввода – вывода $\bar{\tau}_i = \mathcal{G}_i / \bar{n}_i$, $i = 1, \dots, N$.

Модель центрального обслуживания базируется на следующих допущениях: 1) процесс однороден во времени, т.е. этапы ввода – вывода, выполняемые разными устройствами, распределены равномерно во времени продолжительность этапов процесса; 2) процесс обладает марковским свойством, т.е. следующее состояние процесса зависит только от текущего состояния и не связано с предысторией процесса.

Матрица вероятностей переходов (7.13) порождает случайный процесс со следующими характеристиками. Среднее число этапов $\bar{n}_1, \dots, \bar{n}_N$, дисперсии $D[n_1], \dots, D[n_N]$ и распределения $p[n_1], \dots, p[n_N]$

$$\bar{n}_1 = 1/p_0; \quad \bar{n}_i = p_i/p_0; \quad i = 2, \dots, N;$$

$$D[n_1] = (1-p_0)/p_0^2; \quad D[n_i] = p_i(p_i+p_0)/p_0^2;$$

$$p(n_i) = p_0(1-p_0)^{n_i-1}; \quad n_i = 0, 1, 2, \dots; \quad (7.15)$$

$$p(n_i) = p_0 \left[\frac{p_i(1-p_0)}{1-(1-p_0)(1-p_i)} \right]^{n_i}, \quad n_i = 0, 1, 2, \dots$$

и одноименные характеристики числа попаданий n процесса во все невозвратные состояния, т.е. числа этапов, составляющих процесс –

$$\bar{n}_1 = (2 - p_0) / p_0; \quad D[n] = 4(1 - p_0) / p_0$$

$$p(n) = p_0(1 - p_0)^n, \quad n = 3, 5, 7, \dots \quad (7.16)$$

Таким образом, если модель центрального обслуживания применима для представления реальных процессов, то сведения о числе обращений n_2, \dots, n_N к периферийным устройствам R_2, \dots, R_N и длительности процессорной обработки \mathcal{G}_1 и ввода-вывода $\mathcal{G}_2, \dots, \mathcal{G}_N$ достаточны для определения параметров марковского процесса. Более детально можно представить процессы, задаваясь дисперсиями длительности этапов $D^*[\tau_1], \dots, D^*[\tau_N]$ или их распределениями $p^*(\tau_1), \dots, p^*(\tau_N)$.

Однородное и неоднородное представление рабочей нагрузки. Рабочую нагрузку, зафиксированную при измерении процесса функционирования системы в достаточном интервале времени, можно представить среднестатистическим заданием, параметры которого – среднее число обращений $\bar{n}_2^*, \dots, \bar{n}_N^*$ к периферийным устройствам R_2, \dots, R_N и длительностью процессорной обработки и ввода-вывода $\mathcal{G}_1, \dots, \mathcal{G}_N$ – определяются как статистические средние на множестве выполненных заданий. Представление рабочей нагрузки заданием одного типа со среднестатистическими параметрами называется *однородным*.

В подавляющем большинстве случаев рабочая нагрузка состоит из неоднородных заданий, существенно различающихся по объему используемых ресурсов – в десятки и даже сотни раз. Различия в ресурсоемкости учитываются при обработке данных путем разбиения заданий на классы, каждый из которых объединяет задания с примерно одинаковыми свойствами, но существенно отличными от свойств заданий других классов. Классификация заданий используется для создания мультипрограммных смесей, позволяющих равномерно загружать ресурсы и за счет этого повышать производительность системы, а также при назначении заданиям приоритетов, с помощью которых обеспечивается необходимое время ответа, например малое время для коротких заданий.

Представление о неоднородности нагрузки дает распределение (гистограмма) параметров, таких, как суммарное время выполнения заданий, число обращений n_i^* к периферийным устройствам R_i , $i = 2, \dots, N$, и время использования заданием \mathcal{G}_i устройства R_i . Обычно распределение параметров заданий имеет вид, изображенный на рис. 7.19. Представленное распределение является многомодальным, и его можно трактовать как смесь распределений, соответствующих различным классам заданий в рабочей нагрузке. В данном случае можно предполагать существование четырех классов заданий со значениями параметра x , близкими к модам распределения x_1, \dots, x_4 .

Представление рабочей нагрузки в виде совокупности классов называется *неоднородным*. При нем класс характеризуется долей заданий, относящихся к этому классу, и среднестатистическими свойствами задания, определяющими потребность задания в ресурсах системы (память, процессорное время и объем ввода-вывода).

Необходимость неоднородного представления рабочей нагрузки связана, во-первых, с организацией рациональных режимов обработки, т.е. с высокой производительностью

системы и требуемым качеством обслуживания пользователей. Во-вторых, неоднородное представление позволяет более точно идентифицировать нагрузку, например, моделями центрального обслуживания и создавать более информативные модели производительности вычислительных систем.

Классификация рабочей нагрузки. Наиболее существенный момент классификации – выбор признаков, в качестве которых при классификации рабочей нагрузки выступают параметры, характеризующие потребность заданий в ресурсах системы. Набор признаков должен быть достаточным для разделения на классы объектов с различными свойствами (существенными для классификации) и вместе с тем по возможности минимальным, чтобы упростить процесс классификации. При классификации заданий, выполняемых в режиме пакетной обработки, стремятся оптимизировать мультипрограммную смесь путем составления ее из заданий разных классов, создающих в совокупности одинаковую нагрузку на все ресурсы. Поэтому в качестве признаков классификации используются емкость занимаемой оперативной памяти и интенсивность обращений к периферийным ч устройствам – в расчете на один миллион процессорных операций. При классификации заданий, выполняемых в режиме оперативной обработки, стремятся обеспечить в первую очередь, наилучшее время ответа для работ разной продолжительности и поэтому в качестве признаков применяется объем используемых ресурсов.

Существенными для классификации являются параметры с большими коэффициентами вариации (отношением среднего квадратического отклонения к математическому ожиданию), а параметры, коэффициенты вариации которых, определенные на множестве классифицируемых объектов, близки к нулю, исключаются из состава признаков. Если несколько параметров коррелированы (парные коэффициенты корреляции не меньше 0,7), в качестве признака классификации используется только один из них.

Для классификации рабочей нагрузки наиболее широко используются три метода: 1) параметрическая классификация; 2) классификация по ядру нагрузки; 3) автоматическая классификация – кластер-анализ.

Параметрическая классификация основана на так называемых решающих правилах, которые устанавливают области значений параметров, соответствующие каждому классу. Например, могут использоваться следующие решающие правила:

Класс	Емкость памяти, кбайт	Интенсивность ввода-вывода, c^{-1}
A	$x \leq 128$	$z \leq 10$
B	$x \leq 128$	$z > 10$
C	$128 < x \leq 512$	$z \leq 5$
D	$128 < x \leq 512$	$z > 5$
E	$x > 512$	Любая

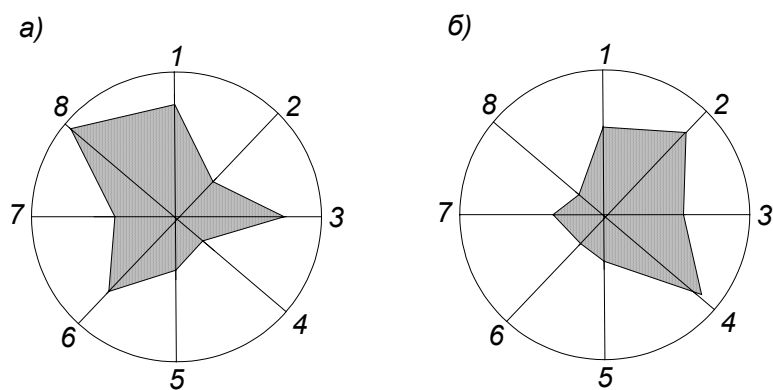


Рис. 7.20. Диаграммы Кивиата для разных заданий

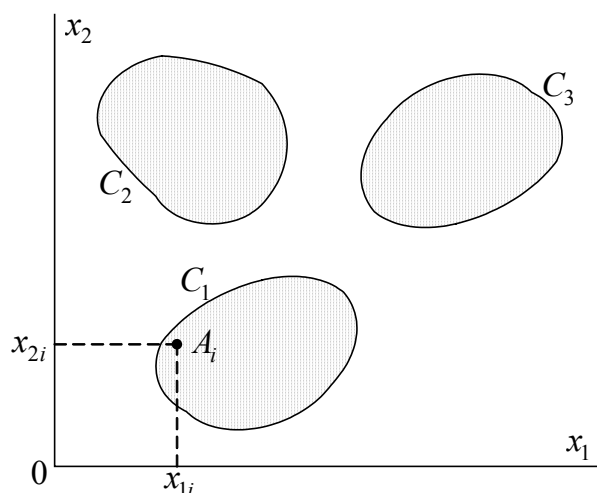


Рис. 7.21. Классификация задач

Согласно этим правилам задание с параметрами $x=200$ и $z=12,5$ будет отнесено к классу D . Решающие правила назначаются исходя из целевых требований к классификации и C состава ресурсов системы. Часто границы классов назначаются на основе многомодальных распределителей значений признаков. При этом в качестве границ принимаются средние точки между соседними модами распределения. Например для распределения на рис. 7.19 могут использоваться следующие правила распределения на классы:

$$x \leq a_1; \quad a_1 < x \leq a_2; \quad a_2 < x \leq a_3; \quad x > a_3$$

Классификация по ядру нагрузки сводится к выделению подмножества заданий, создающих основную нагрузку на систему, например 90% нагрузки. Ядро нагрузки выделяется следующим образом. Для заданий одного наименования определяется число реализаций и средние показатели нагрузки, по которым вычисляется суммарная нагрузка, создаваемая всеми реализациями задания. Задания упорядочиваются по убыванию суммарной нагрузки на ресурсы системы. Первые N заданий, создающие в сумме 90%-ную нагрузку, рассматриваются в качестве ее ядра. Как правило, ядро состоит из небольшого числа заданий, обычно 10–20, и вполне обозримо. Путем анализа параметров заданий, входящих в ядро, назначаются классы каждый из которых состоит из небольшого числа заданий, обычно 1–3. Для выделенных классов устанавливаются граничные значения параметров, на основе которых строятся решающие правила для параметрической классификации заданий, не вошедших в ядро. Во многих случаях последние просто объединяются в один дополнительный класс.

При классификации удобно представлять свойства объектов в виде диаграммы Кивиата (рис. 7.20). Диаграмма состоит из совокупности осей 1, 2, 3, ..., на которых в определенном масштабе в направлении от центра отмечаются значения параметров объекта. Затем точки на осях соединяются прямыми линиями, образующими некоторую фигуру (образ объект a). (Использование диаграмм Кивиата основано на способности человека распознавать сходство и различие геометрических фигур). Если диаграммы, соответствующие заданиям, имеют сходную конфигурацию, задания можно объединить в один класс. Различие конфигураций свидетельствует о несходстве заданий и является основанием для отнесения заданий к разным классам.

Методы *автоматической классификации – кластер-анализа*: – основаны на использовании вычислительных процедур, оценивающих расстояния между классифицируемыми объектами в объединяющих близлежащие объекты в компактные

множества, называемые *кластерами* (классами). При классификации объектов по n признакам каждый объект рассматривается как точка в n -мерном пространстве с координатами, оопределяемыми признаками классификации. На рис. 7.21 приведена типичная картина образования кластеров. Здесь x_1 и x_2 – значения признаков классификации, A_i – объект с признаками (x_{1i}, x_{2i}) и C_1, C_2, C_3 – кластеры, образованные подмножествами классифицируемых объектов. Число кластеров, выявляемых в процессе автоматической классификации, зависит от числа признаков классификации, их дисперсии и свойств объектов. Чем больше число признаков и их дисперсии, тем больше вероятность существования значительного числа кластеров.

В силу автоматизма, присущего всем методам кластер-анализа, классификация ведется вне прямой связи с ее целями. Процесс классификации управляется путем выбора признаков, их масштабирования и задания значений управляющих параметров, используемых процедурами классификации для оценки компактности подмножеств. При классификации десятков или нескольких сотен объектов затраты машинного времени умеренны, а большего числа объектов – весьма значительны. Классификация позволяет представить множество заданий, составляющих рабочую нагрузку, небольшим числом классов заданий, обычно 3–10 классами. За счет этого, описание рабочей нагрузки становится весьма компактным и в то же время сохраняется возможность дифференцировать системные характеристики, характеристики процессов и использования ресурсов по отношению к разнотипным заданиям, связанным с различными группами пользователей.

Системная нагрузка. Выполнение прикладных процессов воздерживается системными процессами. Нагрузка, создаваемая системными процессами, оказывается достаточно большой и составляет десятки процентов для процессора и внешней памяти. Поэтому системная нагрузка учитывается при анализе производительности систем, а также в моделях производительности, используемых для выбора конфигурации и режимов функционирования.

В отличие от рабочей нагрузки, для которой потребность в ресурсах связывается с отдельными заданиями, системная нагрузка представляется как единое целое, относящееся ко всем выполняемым работам. Системная нагрузка регистрируется мониторами как одна из системных характеристик, оцениваемая, например, коэффициентом загрузки процессора со стороны системных процессов. Для определения объема используемых ресурсов необходимо системную нагрузку распределить между всеми заданиями. Аналогично для прогнозирования системной нагрузки при изменении рабочей нагрузки необходимо установить зависимость между ними. Поэтому системную нагрузку стремятся выразить как функцию параметров рабочей нагрузки.

Наиболее широко используется представление системной нагрузки в виде уравнений регрессии. В качестве параметров нагрузки используются емкость памяти, выделяемая для размещения операционной системы, число процессорных операций или коэффициент загрузки процессора системными процессами и аналогичные параметры каналов ввода–вывода. Системная загрузка процессора и каналов ввода–вывода наиболее существенно зависит от следующих параметров заданий: числа шагов и числа операторов языка управления, указанных в задании. При измерениях регистрируются параметры системной нагрузки и одновременно указанные параметры заданий. Затем измерительные данные обрабатываются с помощью программ регрессионного анализа с целью получения уравнений регрессии.

ПРОЕКТИРОВАНИЕ И ЭКСПЛУАТАЦИЯ

8.1. ОРГАНИЗАЦИЯ ПРОЕКТИРОВАНИЯ

Проектирование вычислительных систем – создание комплекта конструкторской и программной документации, необходимой для производства и эксплуатации.

Цель проектирования, назначение проектируемой системы, исходные данные к техническим требованиям устанавливаются техническим заданием. В технических требованиях задаются: функции системы и, возможно, характеристики рабочей нагрузки; состав и характеристики источников и приемников данных (устройств ввода–вывода), а также их территориальное размещение; характеристики системы – производительность, емкость памяти, надежность, стоимость, массогабаритные и энергетические; необходимые режимы функционирования системы (режимы обработки данных); условия эксплуатации и другие факторы, существенные для систем того или иного назначения. В зависимости от предварительной проработки принципов построения технические требования к системе могут задаваться с разной степенью детализации. Функции, возлагаемые на систему, задаются описанием состава задач, для решения которых предназначена система, и при наличии более подробных данных – характеристиками задач, определяющими потребности задач в памяти, процессорной обработке и средствах ввода–вывода. Кроме того, могут быть заданы архитектура системы (ЭВМ, комплекса, сети), технические и программные средства, стадии проектирования.

Стадии проектирования установлены ГОСТ 2.103–68: техническое предложение, эскизный проект, технический проект, рабочая конструкторская документация. Техническое предложение – совокупность конструкторских документов, содержащих техническое и экономическое обоснование проекта как результат анализа различных вариантов построения системы, а патентных исследований. Техническое предложение является основанием для разработки эскизного проекта. Эскизный проект – совокупность конструкторских документов, дающих общее представление о структуре и принципе функционирования проектируемой системы и определяющих основные ее параметры. Эскизный проект является основанием для разработки технического проекта или рабочей документации. Технический проект – совокупность конструкторских документов, содержащих окончательное техническое решение и дающих полное представление о проектируемой системе. На основе технического проекта создается комплект рабочей конструкторской документации, предназначенный для изготовления и испытания опытного образца, а также для производства систем.

Вычислительные системы – сложные системы с иерархической организацией структуры и процессов функционирования. Принято выделять следующие основные уровни, представления структуры и функций: системотехнический, схемотехнический и конструкторский.

На системотехническом уровне структура технических средств определяется с точностью до устройств и интерфейсов. Состав программного обеспечения задается перечнем программ, лингвистического и информационного обеспечения системы (языки программирования, управление данными и заданиями, базы и типы наборов данных). Функционирование системы определяется в терминах процессов, реализация которых связана с использованием памяти, устройств, программ и наборов данных. На схемотехническом уровне определяется структура устройств в форме структурных, функциональных и принципиальных электрических схем, детально представляющих организацию отдельных подсистем и устройств системы. Функционирование устройств

описывается в виде микропрограмм, временных диаграмм, автоматов, булевых функций и электрических процессов. На конструкторском уровне вычислительная система описывается как совокупность конструктивных единиц – шкафов, секций и типовых элементов замены, связанных между собой кабельными, проводными и печатными соединениями. Многоуровневое представление структур и функционирования позволяет описать вычислительную систему с необходимой полнотой и одновременно в наиболее компактной лаконичной форме.

Вычислительные системы проектируются по схеме «сверху вниз» – от верхних уровней представления структуры и функционирования к нижним, т. е. сначала на системотехническом уровне и затем на схемотехническом и конструкторском. В связи с этим выделяются этапы системотехнического, схемотехнического и конструкторского проектирования вычислительных систем. По завершении системотехнического проектирования разрабатывается параллельно со схемотехническим и конструкторским проектированием программное обеспечение. Порядок разработки и состав программной документации определяется стандартами Единой системы программной документации (ЕСПД).

8.2. СИСТЕМОТЕХНИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

Цель системотехнического проектирования – определить следующие аспекты построения вычислительной системы:

- 1) структуру (конфигурацию) – состав технических средств (ЭВМ, устройства, каналы передачи данных) и связи между ними: (система интерфейсов);
- 2) режим функционирования – способы взаимодействия пользователей с системой и организацию вычислительных процессов (ввод, хранение, обработка и вывод данных);
- 3) состав лингвистического, информационного и программного обеспечения и порядок взаимодействия программ;
- 4) характеристики – производительность, время ответа, надежность, стоимость и др.

Структура, режим функционирования, состав лингвистического, информационного и программного обеспечения должны быть оптимально согласованы с назначением системы, определяемым техническим заданием, и в конечном итоге обеспечивать требуемые характеристики системы.

Современное состояние теории вычислительных систем не позволяет решать все задачи системотехнического проектирования формальными методами, устанавливающими последовательность действий для получения оптимального результата. Поэтому существенная роль отводится эвристическим методам решения, основанным на опыте и интуиции проектировщиков. На практике системотехническое проектирование сводится к решению трех основных задач: выбора и разработки базовой структуры, обеспечения требуемой производительности и надежности. Первая задача – главная, так как при ее решении учитывается назначение системы и требования к производительности и надежности. Задачи обеспечения требуемой производительности и надежности направлены на оптимизацию параметров, структуры системы и технических характеристик устройств и интерфейсов в рамках базовой структуры. При решении этих задач, широко используются модели и методы теории вычислительных систем и теории надежности.

Выбор и разработка базовой структуры. Существует широкая номенклатура систем обработки данных: системы централизованной обработки на основе одной или многих несвязанных ЭВМ, обеспеченных при необходимости средствами теледоступа; многомашинные и многопроцессорные комплексы; глобальные (региональные) и локальные вычислительные сети. Перечисленные системы существенно различаются по производительности и надежности, объему хранимых данных и сложности реализуемых алгоритмов, а также в отношении расширения функциональных возможностей, увеличения производительности и надежности. Если в техническом задании не определен тип вычислительной системы, то первая задача системотехнического проектирования – выбор

типа системы, наилучшим образом удовлетворяющего ее назначению. Основные факторы, влияющие на выбор типа системы, – рабочая нагрузка, режим взаимодействия пользователей с системой, требования к надежности и состояние производственно-технической базы, на основе которой предполагается создавать систему.

Рабочая нагрузка определяет потребность задач (пользователей) в ресурсах-памяти, процессорной обработки и ввода – вывода данных. На основе данных о рабочей нагрузке можно приближенно оценить минимальную необходимую емкость оперативной и внешней памяти и тип внешних запоминающих устройств, минимальную необходимую производительность процессоров и пропускную способность подсистемы ввода – вывода. Режим взаимодействия пользователей с системой – пакетная обработка, «запрос – ответ», диалоговый режим и обработка в реальном времени – влияет на уровень загрузки ресурсов системы, состав и число устройств ввода – вывода данных. При пакетной обработке: загрузка процессоров, памяти и системных устройств ввода – вывода оказывается наибольшей и составляет примерно 75–90 %. При обработке данных в режиме «запрос – ответ», диалоговом; режиме и реальном времени для уменьшения времени ответа создается запас производительности устройств обработки и ввода – вывода, а также запас емкости памяти, чтобы сохранить необходимое качество обслуживания при пульсациях нагрузки. За счет этого загрузка основных устройств и памяти системы составляет обычно не более 50–70 %. Таким образом, исходя из сведений о рабочей нагрузке и достижимого уровня загрузки устройств определяется примерная потребность в емкости памяти, производительности процессоров и в средствах ввода – вывода. Кроме того, тип данных, обработка которых возлагается на систему, предопределяет тип операций над ними: целочисленной арифметики, с плавающей запятой, десятичной арифметики над полями переменной длины, обработки текстов, матриц и т. д. Эти факторы влияют на выбор типа процессоров и ЭВМ.

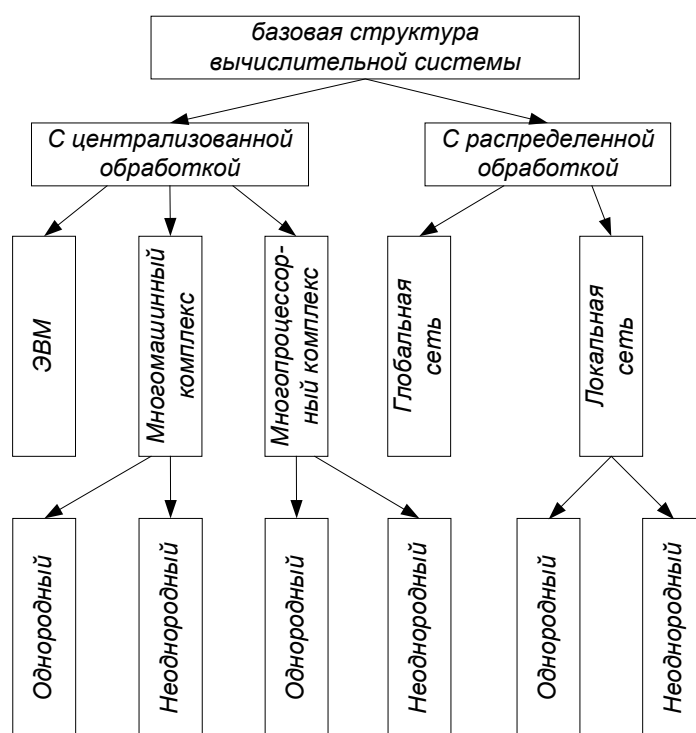


Рис. 8.1. Варианты базовой структуры вычислительных систем

При выборе базовой структуры наиболее сложный момент – анализ состояния производственно-технической базы и возможности использования в проектируемой системе серийно выпускаемых технических средств (ЭВМ, процессоры, внешние запоминающие устройства, устройства ввода–вывода, аппаратура передачи данных и сопряжения каналов с

ЭВМ) и существующего программного обеспечения. При проектировании вычислительных систем, за исключением принципиально новых, стремятся в максимальной степени использовать серийно выпускаемые устройства и существующее программное обеспечение, ориентируясь на средства ЕС ЭВМ, СМ ЭВМ или совместное применение этих средств.

В результате оценки потребностей в ресурсах и анализа состояния производственно-технической базы выбирается и обосновывается базовая структура проектируемой системы. При этом различные варианты построения систем (рис. 8.1) выравниваются по степени соответствия заданной рабочей нагрузке, требованиям к производительности и надежности с учетом состояния производственно-технической базы. В рамках выбранного варианта детализируются существенные моменты структурной организации и функционирования вычислительной системы: для многомашинных комплексов выбираются способы комплектования ЭВМ, разделения, рабочей нагрузки между ЭВМ и управления процессами, для многопроцессорных комплексов – способ доступа к общей памяти и периферийным устройствам, для вычислительных сетей – способ передачи данных и состав функций. В результате исследований, проводимых на стадиях технического предложения и эскизного проекта, формируется базовая структура вычислительной системы, обеспечивающая необходимую производительность процессорной обработки, надежность, а также режимы обслуживания пользователей.

Обеспечение требуемой производительности. Производительность вычислительной системы зависит от следующих основных факторов: 1) структуры системы; 2) режима функционирования; 3) характеристик рабочей нагрузки. Кроме того, на производительность существенно влияет надежность системы, поскольку в результате отказов часть ресурсов или вся система теряет работоспособность на время восстановления системы.

Обеспечение требуемой производительности в процессе системотехнического проектирования сводится к оценке производительности исследуемых вариантов построения системы, а также к оптимизации структуры и режима функционирования для достижения заданной производительности при минимальной стоимости системы.

Основным инструментом при исследовании производительности являются модели производительности вычислительных систем. Создаваемые модели базируются на эмпирических представлениях о процессе функционирования проектируемой системы и, естественно, не могут превосходить их по сложности. Структура проектируемой системы достаточно полно и точно задается выбранной базовой структурой и воспроизводится в модели с любой необходимой степенью подробности. Однако, формирование представлений о рабочей нагрузке (создаваемой потоками обрабатываемых задач) и системной нагрузке (порождаемой операционной системой при заданном режиме обработки задач и составе рабочей нагрузки) вызывает существенные трудности, обусловленные следующими причинами.

Сведения о рабочей и системной нагрузке проектируемых систем, за исключением специализированных систем с ограниченным и фиксированным набором функций, обычно получают путем измерения нагрузки систем, находящихся в эксплуатации и обрабатывающих задачи подобного типа. Но даже при наличии представительной статистики, о работе существующих систем перенос имеющихся данных на систему с другой структурой и операционной системой приводит к значительным погрешностям в прогнозировании нагрузки, что связано со следующими обстоятельствами. Во-первых, при различии структуры и характеристик могут измениться принципы обработки данных и режим взаимодействия пользователей с проектируемой системой и, как следствие, характеристики рабочей нагрузки. Так, использование в проектируемой системе оперативной памяти большой емкости может привести к изменению структуры программ и перемещению в оперативную память значительной части операционной системы и данных, в результате чего уменьшится интенсивность обращений к внешней памяти. Использование во внешней памяти проектируемой системы накопителей большой емкости изменяет порядок размещения наборов данных по накопителям и интенсивность обращения к ним. Увеличение

числа терминалов приводит к уменьшению объема ввода – вывода данных через системные устройства и т. д.

Во-вторых, существенно влияет на рабочую и системную нагрузку различие операционных систем в проектируемой и существующих вычислительных системах, так как каждая операционная система специфично использует ресурсы оперативной памяти, процессоры и внешнюю память, создавая присущую ей системную нагрузку.

В-третьих, наличие средств телеобработки, возможность распределенной обработки в рамках вычислительной сети, организация данных в виде баз и другие подобные аспекты организации проектируемой системы, отсутствующие в существующих системах, влияют на процессы обработки данных и, следовательно, на характеристики нагрузки.

Таким образом, чем больше отличие проектируемой системы от существующих, тем меньше вероятность корректного воспроизведения рабочей и системной нагрузки в моделях производительности.

Модели производительности строятся с учетом уровня достоверности данных о рабочей и системной нагрузке, которыми располагают проектировщики. При невысокой достоверности данных используются наиболее простые модели и приближенные методы расчета их характеристик. С помощью моделей оцениваются следующие характеристики системы:

- 1) загрузка ресурсов и при необходимости профиль загрузки;
- 2) профили процессов, определяющие время пребывания заданий на разных стадиях их выполнения и в очередях;
- 3) производительность.

Модели должны воспроизводить следующее:

- 1) состав и технические характеристики основных устройств системы – процессоров, оперативной памяти, внешних запоминающих устройств и, возможно, каналов ввода – вывода и передачи данных;
- 2) основные параметры режима функционирования – уровень мультипрограммирования и системную нагрузку;
- 3) параметры нагрузки в однородном представлении или в виде нескольких классов задач, обрабатываемых, например, в пакетном режиме, двух-трех классов задач, обрабатываемых в оперативном режиме.

На стадиях технического предложения и эскизного проектирования, когда основное внимание уделяется выбору структуры системы и самых существенных параметров режима функционирования, погрешность оценки характеристик моделируемой системы на уровне 25–50% считается вполне, удовлетворительной. Поэтому для оценки характеристик проектируемых систем используются наиболее простые аналитические методы и несложные имитационные модели. При этом параметры структуры, режима функционирования и нагрузки воспроизводятся в моделях на уровне средних и редко на уровне средних и дисперсий. При этом, как правило, используется комплекс моделей производительности: общая модель производительности системы в целом и частные модели, воспроизводящие функционирование отдельных подсистем – общей памяти с многоканальным доступом, моноканала, виртуальной памяти и др. Частные модели позволяют оценить характеристики отдельных процессов, существенно влияющих на производительность системы, например вероятность блокировки обращений при доступе к общим модулям оперативной памяти и внешним устройствам, задержку передачи данных через общую шину, частоту страничных сбоев в виртуальной памяти и т. п. Характеристики, полученные с помощью частных моделей, вводятся в общую модель производительности в качестве параметров, представляющих свойства процессов более низких уровней (по сравнению с уровнями процессов, воспроизводимых общей моделью).

Оптимизация проектируемой системы по производительности сводится к согласованию структуры системы и технических характеристик устройств с режимом функционирования системы и рабочей нагрузкой. Согласование означает в основном

выявление и устранение узких мест в системе. Узкое место – это ресурс (устройство или память), нехватка которого приводит к недоиспользованию остальных ресурсов системы. Устраняется узкое место путем изменения числа или характеристик соответствующих устройств. Если улучшить характеристики этого ресурса невозможно, изменяют структуру и режим функционирования системы. Например, невозможность увеличения производительности процессора может потребовать многомашинной организация системы, в частности использования сателлитной ЭВМ. При этом с помощью моделей производительности оценивается эффективность различных вариантов построения системы и выбирается вариант, в наибольшей степени соответствующий назначению проектируемой системы.

В результате исследования производительности на стадиях технического предложения и эскизного проекта базовая структура проектируемой системы конкретизируется до значений параметров, определяющих число и технические характеристики устройств, пропускную способность интерфейсов и каналов связи.

Модели производительности необходимы при эскизном и техническом проектировании программного обеспечения для оценки эффективности различных вариантов организации вычислительных процессов. Здесь требуются более детальные модели, чем при разработке базовой структуры системы. В моделях необходимо воспроизводить процессы планирования и выполнения работ, реализуемые управляющими программами операционной системы, и рабочую нагрузку в неоднородном представлении – в виде совокупности классов заданий, обрабатываемых в разных режимах, имеющих приоритеты и использующих различный состав и объем ресурсов. Наиболее подходящими для этой цели являются гибридные модели, в которых рабочая нагрузка и процессы планирования воспроизводятся имитационными, а процессы выполнения задач – аналитическими методами. Имитационные методы позволяют моделировать сколь угодно сложную организацию планирования и выполнения задач и используются для построения моделей производительности, детально отображающих все аспекты функционирования проектируемой системы.

Обеспечение надежности. Требования к надежности определяются назначением вычислительной системы и задаются, как правило, минимальным, допустимым: коэффициентом готовности, характеризующим долю времени, в течение которого система сохраняет работоспособность. В общем случае коэффициент готовности $K_r = T / (T + T_g)$, где T – средняя наработка на отказ и T_g – среднее время восстановления. Для систем повышенной надежности, которые должны функционировать непрерывно, требование к надёжности может характеризоваться предельным допустимым числом часов простоя системы в заданный период времени, например 2 часа за 10 лет работы. В тех случаях, когда некоторые группы отказов приводят к прекращению обслуживания отдельных пользователей при сохранении работоспособности остальных, требования к надежности задаются отдельно для подсистем вычислительной системы, обслуживающих часть рабочей нагрузки, и для ядра системы.

Обеспечение надежности системы охватывает все стадии проектирования, производства и эксплуатации системы. Совокупность мероприятий, направленных на повышение надежности системы, оформляются в виде программы обеспечения надежности, в которой предусматриваются необходимые мероприятия, гарантирующие надежность технических и программных средств.

На стадии технического предложения и эскизного проекта оценивается надежность базовой структуры вычислительной системы. Вычисленные показатели сравниваются с требуемыми, и определяется способ повышения надежности системы. Надежность можно увеличить за счет использования высоконадежных элементов, облегчения режима работы элементов и совершенствования технологии изготовления и сборки системы. Другой подход к повышению надежности – обеспечение отказоустойчивости системы за счет резервирования. Резервировать можно технические средства на уровне логических элементов, устройств и ЭВМ или программные средства путем повторной реализации

функций на основе одного и того же или разных алгоритмов и создания дополнительных программ для выявления и исправления ошибок. При резервировании технических средств в систему вводится аппаратный резерв, при резервировании программных средств – временной резерв. Резервирование технических средств может выполняться различными способами – статическим, динамическим и гибридным. При статическом резервировании отказы блокируются автоматически, без изменения структуры и режима функционирования системы. Наиболее распространено тройное статическое резервирование – с использованием схемы голосования (мажоритарной латки), которая формирует результат, соответствующий совпадающим результатам двух из трех модулей, вследствие чего отказ одного модуля не нарушает работоспособность системы. При динамическом резервировании модули оснащаются средствами, контролирующими их работоспособность, и при обнаружении отказа производится реконфигурация системы для подключения резервного модуля или перераспределения функций между исправными модулями системы. Гибридное резервирование основано на совместном использовании принципов статического и динамического резервирования.

Эффективность вариантов повышения надежности оценивается с помощью моделей надежности, создаваемых на основе аппарата теории надежности. Определяются показатели надежности, и прирост надежности сопоставляется с затратами на введение резерва в систему. Путем сравнения вариантов выбирается наиболее, рациональный. Значительные трудности возникают при разделении требований к надежности между частями системы – ее подсистемами и устройствами. Чтобы обеспечить заданную надежность при минимальных затратах, необходимо установить соотношение между надежностью и затратами на ее обеспечение, для каждого типа устройств и подсистем. Оценка затрат производится на основе опыта предыдущих разработок, причем, естественно, с рядом допущений. По этой причине устанавливать требования к надежности подсистем приходится в условиях существенной неопределенности.

Для вычислительных систем характерны сбои в работе, вызванные электромагнитными помехами, пульсацией питания и другими причинами. Действие этих факторов обычно кратковременно. Для обеспечения работоспособности в условиях сбоев в систему встраиваются средства, позволяющие обнаружить ошибки и запускать процедуру восстановления (например, повторение действий). Если процедура восстановления работоспособности выполнена успешно, то причиной ошибки считается сбой и система продолжает функционировать в естественном режиме. В противном случае отказ считается катастрофическим.

В программе обеспечения надежности системы важная роль отводится повышению ремонтпригодности, позволяющей снижать затраты времени и средств на восстановление работоспособности системы. Для повышения ремонтпригодности существует множество способов, реализуемых на всех стадиях проектирования, производства и эксплуатации системы: встраивание в систему эффективных средств контроля и диагностики, унификация конструкций и рациональная их компоновка и др.

8.3. ЭКСПЛУАТАЦИЯ

Эксплуатация вычислительной системы (ЭВМ, комплекса или сети) состоит из технического и системотехнического обслуживания системы и использования ее по прямому назначению – для обработки данных.

Техническое обслуживание (техническая эксплуатация) – обеспечение работоспособности системы путем создания требуемых условий эксплуатации (режим электропитания, температурный и др.) и проведения профилактических и ремонтно-восстановительных работ. Основной показатель качества технического обслуживания – коэффициент готовности системы (компонентов), характеризующий долю времени, в течение которого система (компоненты) работоспособна. Увеличение коэффициента готовности системы достигается путем сокращения простоев оборудования из-за

профилактических и ремонтно-восстановительных работ. На эффективность технического обслуживания наиболее существенно влияют следующие факторы:

- 1) условия эксплуатации системы (стабильность электропитания, температура, чистота среды и др.);
- 2) надежность и ремонтпригодность системы, степень совершенства средств контроля и диагностики;
- 3) режим обслуживания и квалификация обслуживающего персонала;
- 4) полнота данных о сбоях и отказах технических средств.

Для повышения эффективности технического обслуживания в системах предусматриваются средства накопления данных об ошибках, сбоях и отказах. Данные накапливаются путем регистрации состояния системы в момент ошибки, обнаруживаемой встроенными средствами контроля или программами. Регистрация данных производится операционной системой в специальном системном журнале – области накопителя на магнитных дисках. Данные из системного журнала периодически или при необходимости выводятся на печать и используются обслуживающим персоналом для выявления источников ошибок, сбоев и отказов с целью проведения профилактических и ремонтно-восстановительных работ.

Системотехническое обслуживание (системотехническая эксплуатация) – обеспечение эффективности использования системы, направленное на снижение стоимости обработки данных, повышение производительности системы, качества обслуживания пользователей и др.

Основные задачи системотехнического обслуживания:

- 1) выбор и адаптация операционных систем, в том числе общесистемного программного обеспечения;
- 2) настройка операционной системы на рабочую нагрузку – организация и выбор параметров функционирования системы, обеспечивающих требуемое качество обслуживания пользователей и максимальную производительность;
- 3) совершенствование конфигурации системы – состава устройства и связей между ними.

Все задачи системотехнической эксплуатации сводятся к взаимному согласованию конфигурации, режима функционирования системы и рабочей нагрузки для обеспечения требуемого качества обслуживания пользователей – организации необходимых режимов обработки данных, увеличения производительности, уменьшения времени ответа и стоимости обработки данных.

Выбор операционной системы производится исходя из конфигурации вычислительной системы (производительность процессора, ёмкость оперативной памяти, состав внешней памяти и средств ввода – вывода), потребности в режимах обработки данных (сосредоточенная, телекоммуникационная, сетевая, пакетная, диалоговая и т. д.) и основных свойств рабочей нагрузки (состав и характеристики решаемых системой задач). Операционная система адаптируется к конфигурации вычислительной системы и потребностям пользователей путем генерации варианта, который должен содержать необходимые средства управления устройствами и памятью, способы доступа к данным и обеспечивать требуемые режимы обработки данных. При этом оперативная и внешняя память разделяется на области, предоставляемые системному и прикладному программному обеспечению. В состав общесистемного программного обеспечения включаются необходимые средства автоматизации программирования, отработки символьной и графической информации, управления базами данных, телеобработки и т. д.

Полученный вариант операционной системы определяет возможные режимы функционирования вычислительной системы и обработки данных с точностью до значений параметров, устанавливающих уровень мультипрограммирования, величину кванта процессорного времени, число системных процессов ввода – вывода и другие показатели,

которые задаются и изменяются в процессе эксплуатации вычислительной системы путем настройки операционной системы на действующую рабочую нагрузку.

При генерации операционной системы рабочая нагрузка может быть предсказана лишь приблизительно. К тому же она меняется во времени. Поэтому настройка операционной системы на реальную рабочую нагрузку проводится на работающей вычислительной системе и повторяется неоднократно. Цель настройки – повысить производительность системы и обеспечить требуемую оперативность обработки – необходимое время ответа. Настройка сводится к назначению параметров операционной системы: уровня мультипрограммирования, кванта процессорного времени, размеров буферов, алгоритмов планирования заданий и задач, алгоритмов управления памятью и устройствами т. д.

Необходимая для настройки операционной системы информация собирается при эксплуатации вычислительной системы с помощью системы оценки функционирования, содержащей мониторы и средства обработки измерительных данных. Функционирование вычислительной системы оценивается совокупностью следующих характеристик:

- 1) ресурсоемкости выполненных работ, рабочей и системной нагрузки;
- 2) загрузки ресурсов системы и структурной загрузки;
- 3) общесистемных – производительности, времени ответа и профилей процессов.

Путем анализа зарегистрированных Характеристик выявляются перегруженные и недоиспользуемые ресурсы системы и узкие места, негативно влияющие на производительность а время ответа. С помощью настроечных параметров операционной системы можно изменить уровень и структуру загрузки ресурсов профили процессов и, следовательно, производительность и время ответа. Эффект, достигаемый за счет изменения настроечных параметров, прогнозируется либо эвристически, на основе априорных представлений о степени влияния параметров на характеристики системы, либо с помощью моделей производительности.

Существенное влияние на производительность и время ответа оказывает способ размещения информации в памяти системы. Так, интенсивность обращения к внешней памяти сильно зависит от того, какие модули операционной системы объявляются резидентными и размещаются в оперативной памяти. Время доступа к накопителям на магнитных дисках существенно зависит от порядка размещения наборов данных по накопителям и в пределах каждого накопителя. С помощью мониторов определяется интенсивность обращений к каждому из накопителей (тому данных) и набору данных, а также распределение интенсивности обращений по цилиндрам накопителей. За счет перераспределения данных по томам обеспечивается равномерная загрузка внешних запоминающих устройств, а следовательно, уменьшается время доступа к внешней памяти. За счет перемещения наборов данных в пределах каждого тома уменьшается время доступа к накопителям [1, 18].

Систематическое проведение работ по измерению и анализу функционирования вычислительной системы и настройке операционной системы позволяет оптимизировать значения настроечных параметров. В таком случае дальнейшая настройка не может дать существенных улучшений и для повышения производительности необходимо совершенствовать конфигурацию системы: увеличивать число устройств или использовать устройства с более высокими характеристиками.

Совершенствование конфигурации вычислительной системы проводится аналогично настройке операционной системы. С помощью измерений получают необходимые данные о функционировании вычислительной системы, в частности, данные о загрузке устройств. Если загрузка отдельных устройств или групп относительных устройств близка к предельной и настройка операционной системы не способна изменить загрузку, необходимо увеличивать число соответствующих устройств, например накопителей на магнитных дисках, каналов связи, или применять устройства с лучшими характеристиками. Для того чтобы оценить эффект, получаемый за счет совершенствования конфигурации системы, используются

модели производительности, с помощью которых прогнозируются характеристики системы и выявляется оптимальный вариант развития конфигурации. Изменение конфигурации влечет необходимость проведения работ по настройке операционной системы на новую конфигурацию и рабочую нагрузку.

Возможности эволюционного развития конфигурации и совершенствования режима обработки для удовлетворения меняющихся требований могут оказаться исчерпанными. Тогда возникает необходимость в качественных изменениях – полной замене ЭВМ на более производительную или предпочтительную по другим характеристикам. Данная задача в практике систематической эксплуатации называется задачей выбора.

В строгой постановке задача выбора является сложной многопараметрической и многокритериальной задачей оптимизации, методы решения которой далеки от завершенности. Практически задача решается в условиях выбора из некоторого числа промышленно освоенных или планируемых к выпуску систем. Основу выбора составляют ресурсные оценки и опыт использования имеющихся средств.

Для выбора должны быть получены следующие данные:

- 1) об использовании имеющихся ресурсов задачами различных классов;
- 2) об использовании ресурсов для обеспечения различных режимов обработки (пакетного, оперативного)

Прогноз изменений объема и состава нагрузки, режимов обработки и требований к обслуживанию должен основываться на соответствующих организационных и технико-экономических предпосылках.

Сложность прогнозирования состоит в том, что для получения надежного прогноза необходимо располагать данными о ресурсоемкости различных классов задач и их влиянии на показатели использования ЭВМ. Для получения таких данных необходим анализ всего процесса использования ЭВМ в конкретной сфере применения. Прогноз потребностей в вычислительных ресурсах связывается с планово-экономическими данными, например с планируемым изменением объема выпускаемой продукции или объема проектных и других работ. Результаты прогнозирования должны определять требуемую интенсивность обслуживания для основных типов ресурсов (процессора и устройства ввода-вывода) по каждому классу задач и режиму обработки.

Вторым этапом выбора является сопоставление прогнозируемых потребностей в ресурсах с характеристиками потенциально возможных конфигураций. Первоначально выбирается производительность процессора, которая должна быть достаточной для обработки потока задач. Оценка производительности получается суммированием интенсивности обслуживания по различным классам задач для пакетного и оперативного режима и операционной системы. На основе полученной оценки выбирается номинальная производительность процессора.

Выбор конфигурации системы ввода-вывода состоит в определении состава периферийных устройств, включая канальное оборудование, достаточного для прогнозируемой нагрузки. Необходимо, чтобы пропускная способность ввода-вывода, главным образом дисковой памяти, соответствовала прогнозируемой интенсивности ввода-вывода с учетом ограничений на время ответа для оперативного режима.

СПИСОК ЛИТЕРАТУРЫ

1. Авен О.И., Гурин Н.Н., Коган Я.А. Оценка качества и оптимизация вычислительных систем. М.: Наука, 1982.
2. Боккер П. Передача данных (Техника связи в системах телеобработки данных); Пер. с нем. М.; Радио и связь, 1981. Т. 1, 2.
3. Вейцман К. Распределенные системы мини – и микро-ЭВМ; Пер. с англ. М.: Финансы и статистика, 1982.
4. Вычислительные сети и сетевые протоколы/ Д. Дэвис, Д. Барбер, У. Прайс, С. Соломонидес; Пер. с англ. М.: Мир, 1982.
5. Головкин Б.А. Параллельные вычислительные системы. М.Наука, 1980.
6. Дроздов Е.А., Комарницкий В.А., Пятибратов А.П. Электронные вычислительные машины Единой системы. – 2-е изд. М.: Машиностроение, 1981.
7. Каган Б.М. Электронные машины и системы: Учебное пособие. – 2-е изд., перераб. И доп. М.: Энергоатомиздат, 1985.
8. Кемени Дж., Снелл Дж. Конечные цепи Маркова: Пер. с англ. М.: Наука, 1970.
9. Клейнрок Л. Вычислительные системы с очередями: Пер. с англ. М.: Мир, 1979.
10. Клейнрок Л. Теория массового обслуживания: Пер. с англ. М.: Машиностроение, 1979.
11. Мультипроцессорные системы и параллельные вычисления: Пер. с англ. / Под ред. Ф.Г. Энслоу. М.: Мир, 1976.
12. Основы теории вычислительных систем: Учебное пособие / Под ред. С.А. Майорова. М.: Высшая школа, 1978.
13. Позин И.Л., Щербо В.К. Телеобработка данных в автоматизированных системах. М.: Статистика, 1976.
14. Пospelов Д.А. Введение в теорию вычислительных систем. М.: Сов. Радио, 1972.
15. Прангишвили И.В., Подлазов В.С., Стецюра Г.Г. Локальные микропроцессорные вычислительные сети.. М.: Наука, 1984.
16. Праневичус Г.И. Модели и методы исследования вычислительных систем. Вильнюс: Мокслас, 1982.
17. Сипсер Р. Архитектура связи в распределенных системах: Пер. с англ. М.: Мир, 1981. Т. 1, 2.
18. Феррари Д. Оценка производительности вычислительных систем: Пер. с англ. М.: Мир, 1981.
19. Якубайтис Э.А. Информационно-вычислительные сети. И.: Финансы и статистика, 1984.