

14.1 Методы управления вводом/выводом

В ВМ находят применение три способа организации ввода/вывода (В/ВЫВ):

- программно управляемый ввод/вывод;
- ввод/вывод по прерываниям;
- прямой доступ к памяти.

При *программно управляемом вводе/выводе* все связанные с этим действия происходят по инициативе центрального процессора и под его полным контролем. ЦП выполняет программу, которая обеспечивает прямое управление процессом ввода/вывода, включая проверку состояния устройства, выдачу команд ввода или вывода. Выдав в МВВ команду, центральный процессор должен ожидать завершения ее выполнения, и, поскольку ЦП работает быстрее, чем МВВ, это приводит к потере времени.

Ввод/вывод по прерываниям во многом совпадает с программно управляемым методом. Отличие состоит в том, что после выдачи команды ввода/вывода ЦП должен циклически опрашивать МВВ для выяснения состояния устройства. Вместо этого процессор может продолжать выполнение других команд до тех пор, пока не получит запрос прерывания от МВВ, извещающим о завершении выполнения ранее выданной команды В/ВЫВ. Как и при программно управляемом В/ВЫВ, ЦП отвечает за извлечение данных из памяти (при выводе) и запись данных в память (при вводе).

Повышение как скорости В/ВЫВ, так и эффективности использования ЦП обеспечивает третий способ В/ВЫВ — *прямой доступ к памяти* (ПДП). В этом режиме основная память и модуль ввода/вывода обмениваются информацией напрямую, минуя процессор.

14.2 Программно управляемый ввод/вывод

Наиболее простым методом управления вводом/выводом является *программно управляемый ввод/вывод*, часто называемый также *вводом/выводом с опросом*. Здесь ввод/вывод происходит под полным контролем центрального процессора и реализуется специальной процедурой ввода/вывода. В этой процедуре ЦП с помощью команды ввода/вывода сообщает модулю ввода/вывода, а через него и внешнему устройству о предстоящей операции. Адрес модуля и ВУ, к которому производится обращение, указывается в адресной части команды ввода или вывода. Модуль исполняет затребованное действие, после чего устанавливает в единицу соответствующий бит в своем регистре состояния. Ничего другого, чтобы уведомить ЦП, модуль не предпринимает. Следовательно, для определения момента завершения операции или пересылки очередного элемента блока данных процессор должен периодически опрашивать и анализировать содержимое регистра состояния МВВ.

Если к МВВ подключено несколько ВУ, то в процедуре ввода/вывода

нужно производить циклический опрос всех устройств, с которыми в данный момент производятся операции В/ВЫВ.

Из блок-схемы (см. рис. 14.2.1) явно виден основной недостаток программно управляемого В/ВЫВ — неэффективное использование процессора из-за ожидания готовности очередной порции информации, в течение которого никаких иных полезных действий ЦП не выполняет. Кроме того, пересылка даже одного слова требует выполнения нескольких команд. ЦП должен тратить время на анализ битов состояния MBV, запись в MBV битов управления, чтение или запись данных со скоростью, определяемой внешним устройством. Все это также отрицательно сказывается на эффективности ввода/вывода.

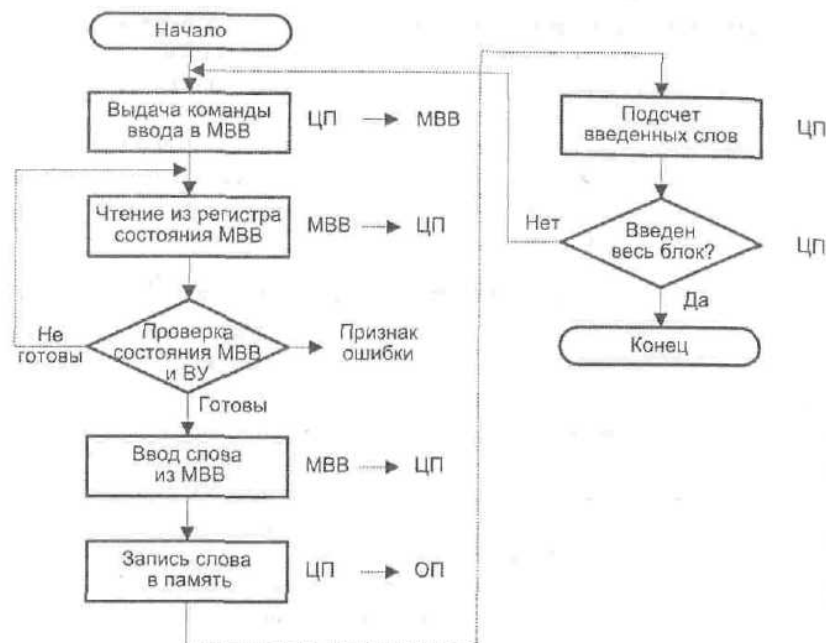


Рис. 14.2.1. Программно управляемый ввод данных

Главным аргументом в пользу программно управляемого ввода/вывода является простота MBV, поскольку основные функции по управлению В/ВЫВ берет на себя процессор. При одновременной работе с несколькими ВУ приоритет устройств легко изменить программными средствами (последовательностью опроса). Наконец, подключение к СВВ новых внешних устройств или отключение ранее подключенных также реализуется без особых сложностей.

14.3 Ввод/вывод по прерываниям

Основным недостатком программно управляемого В/ВЫВ являются простои процессора в ожидании, пока модуль ввода/вывода выполнит очередную операцию. Альтернативой может быть вариант, когда ЦП выдает команду В/ВЫВ, а затем продолжает делать другую полезную работу. Когда ВУ готово к обмену данными, оно через MBV извещает об этом процессор с помощью запроса на прерывание. ЦП осуществляет передачу очередного элемента данных, после чего возобновляет выполнение прерванной программы.

Этот метод эффективнее программно управляемого В/ВЫВ, поскольку устраняет ненужные ожидания, однако обработка прерывания занимает достаточно много времени ЦП. Кроме того, каждое слово, пересылаемое из памяти в модуль В/ВЫВ или в противоположном направлении, как и при программно управляемом В/ВЫВ, проходит через ЦП.

При реализации ввода/вывода по прерываниям необходимо дать ответы два вопроса. Во-первых, определить, каким образом ЦП может выяснить, какой из МВВ и какое из подключенных к этому модулю внешних устройств выставили запрос. Во-вторых, при множественных прерываниях требуется решить, какое из них должно быть обслужено в первую очередь.

Сначала рассмотрим вопрос идентификации устройства. Здесь возможны три основных метода:

- множественные линии прерывания;
- программная идентификация;
- векторное прерывание.

Наиболее простой подход к решению проблемы определения источника запроса — применение *множественных линий прерывания* между ЦП и модулями ввода/вывода.

При *программной идентификации*, обнаружив запрос прерывания, ЦП переходит к общей программе обработки прерывания, задачей которой является опрос всех МВВ с целью определения источника запроса. Для этого может быть выделена специальная командная линия опроса. ЦП помещает на адресную шину адрес опрашиваемого МВВ и формирует на этой линии сигнал опроса. Реакция модуля зависит от того, выставил он запрос или нет. Недостаток метода программной идентификации заключается в больших временных потерях.

Наиболее эффективную процедуру идентификации источника прерывания обеспечивают аппаратные методы, в основе которых лежит идея *векторного прерывания*. В этом случае, получив подтверждение прерывания от процессора, выставившее запрос устройство выдает на шину данных специальное слово, называемое *вектором прерывания*. Слово содержит либо адрес МВВ, либо какой-нибудь другой уникальный идентификатор, который ЦП интерпретирует как указатель на соответствующую программу обработки прерывания. Такой подход устраняет необходимость в предварительных действиях с целью определения источника запроса прерывания. Реализуется он с помощью хранящейся в ОП *таблицы векторов прерывания* (рис. 14.3.1), где содержатся адреса программ обработки прерываний. Входом в таблицу служит вектор прерывания. Начальный адрес таблицы обычно задается неявно, то есть под таблицу отводится вполне определенная часть памяти.

Наиболее распространены два варианта векторной идентификации источник запроса прерывания: цепочечный опрос и арбитраж шины.

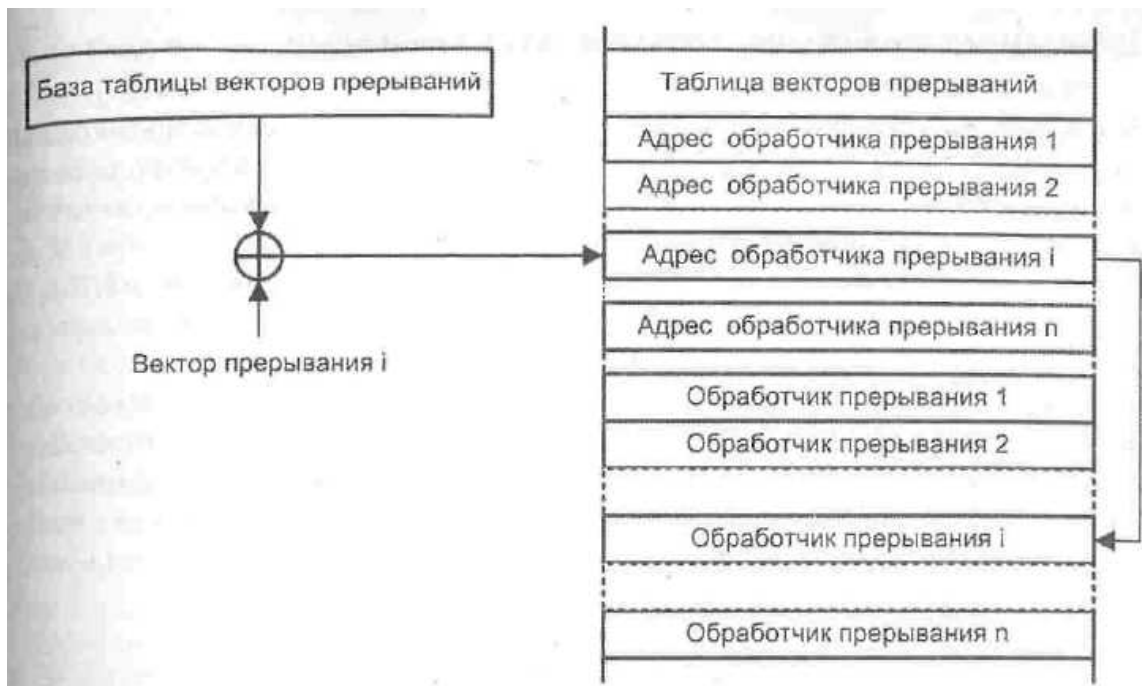


Рис. 14.3.1 Идентификация запроса с помощью вектора прерывания

При *цепочечном методе* для передачи запроса прерывания модули ввода/вывода совместно используют одну общую линию. Линия подтверждения прерывания последовательно проходит через все MBV. Когда ЦП обнаруживает запрос прерывания, он посылает сигнал по линии подтверждения прерывания. Этот сигнал движется через цепочку модулей, пока не достигнет того, который выставил запрос. Запросивший модуль реагирует путем выдачи на шину данных своего вектора прерывания.

В варианте *арбитража шины MBV*, прежде чем выставить запрос на линии запроса прерывания, должен получить право на управление шиной. Таким образом, в каждый момент времени активизировать линию запроса прерывания может только один из модулей. Когда ЦП обнаруживает прерывание, он отвечает по линии подтверждения. После этого запросивший модуль помещает на шину данных свой вектор прерывания.

Перечисленные методы служат не только для идентификации запросившего MBV, но и для назначения приоритетов, когда прерывание запрашивают несколько устройств. При множественных линиях запроса ЦП начинает с линии, имеющей наивысший приоритет. В варианте программной идентификации приоритет модулей определяется очередностью их проверки. Для цепочечного метода приоритет модулей определяется порядком их следования в цепочке.

14.4 Прямой доступ к памяти

Хотя ввод/вывод по прерываниям эффективнее программно управляемого, оба этих метода страдают двумя недостатками:

- темп передачи при вводе/выводе ограничен скоростью, с которой ЦП в состоянии опросить и обслужить устройство;
- ЦП вовлечен в управление передачей, для каждой пересылки он

должен вы полнить определенное количество команд.

Когда пересылаются большие объемы данных, требуется более эффективный способ ввода/вывода — *прямой доступ к памяти* (ПДП). ПДП предполагает наличие на системной шине дополнительного модуля — *контроллера прямого доступа к памяти* (КПДП), способного брать на себя функции ЦП по управлению системной шиной и обеспечивать прямую пересылку информации между ОП и ВУ без участия центрального процессора. В сущности, КПДП — это и есть модуль ввода/вывода, реализующий режим прямого доступа к памяти.

Если ЦП желает прочитать или записать блок данных, он прежде всего должен поместить в КПДП информацию, характеризующую предстоящее действие. Этот процесс называется инициализацией КПДП и включает в себя занесение в контроллер следующих четырех параметров:

- вида запроса (чтение или запись);
- адреса устройства ввода/вывода;
- адреса начальной ячейки блока памяти, откуда будет извлекаться или куда будет вводиться информация;
- количества слов, подлежащих чтению или записи.

Эффективность ПДП зависит от того, каким образом реализовано распределение системной шины между ЦП и КПДП в процессе пересылки блока. Здесь может применяться один из трех режимов:

- блочная пересылка;
- пропуск цикла;
- прозрачный режим.

При *блочной пересылке* КПДП полностью захватывает системную шину с момента начала пересылки и до момента завершения передачи всего блока. На весь этот период ЦП не имеет доступа к шине.

В режиме *пропуска цикла* КПДП после передачи каждого слова на один цикл шины освобождает системную шину, предоставляя ее на это время процессору. Поскольку КПДП все равно должен ждать готовности ПУ, это позволяет ЦП эффективно распорядиться данным обстоятельством.

В *прозрачном режиме* КПДП имеет доступ к системной шине только в тех циклах, когда ЦП в ней не нуждается. Это обеспечивает наиболее эффективную работу процессора, но может существенно замедлять операцию пересылки блока данных. Здесь многое зависит от решаемой задачи, поскольку именно она определяет интенсивность использования шины процессором.

Механизм ПДП может быть реализован различными путями. Некоторые возможности показаны на рис. 14.4.1.

В первом примере (см. рис. 14.4.1, *а*) все ВУ совместно используют общую системную шину. КПДП работает как заменитель ЦП и обмен данными между памятью и ВУ через КПДП производит через программно управляемый ввод/вывод. Хотя этот вариант может быть достаточно дешевым, эффективность его невысока. Как и в случае программно управляемого ввода/вывода, осуществляемого процессором, каждая

пересылка требует двух циклов шины.

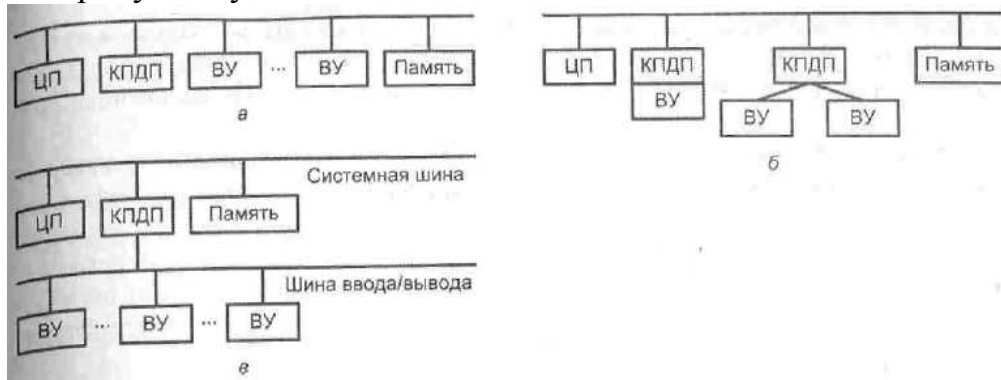


Рис. 14.4.1. Возможные конфигурации систем прямого доступа к памяти

Число необходимых циклов шины может быть уменьшено при объединении функции КПДП и ВУ. Как видно из рис. 14.4.1, б, это означает, что между КПДП и одним или несколькими ВУ есть другой тракт, не включающий системную шину. Логика ПДП может быть частью ВУ, либо это может быть отдельный КПДП, управляющий одним или несколькими внешними устройствами. Допустим и еще один шаг в том же направлении (см. рис. 14.4.1, в) — соединение КПДП с ВУ посредством шины ввода/вывода. Это уменьшает число интерфейсов В/ВЫВ в КПДП и делает такую конфигурацию легко расширяемой. В двух последних вариантах системная шина задействуется КПДП только для обмена данными с памятью. Обмен данными между КПДП и ВУ реализуется минуя системную шину.

14.5 Каналы и процессоры ввода/вывода

По мере развития систем В/ВЫВ их функции усложняются. Главная цель такого усложнения — максимальное высвобождение ЦП от управления процессами ввода/вывода. Некоторые пути решения этой задачи уже были рассмотрены. Следующими шагами в преодолении проблемы могут быть:

1. Расширение возможностей МВВ и предоставление ему прав процессора со специализированным набором команд, ориентированных на операции ввода/вывода. ЦП дает указание такому процессору В/ВЫВ выполнить хранящуюся в памяти ВМ программу ввода/вывода. Процессор В/ВЫВ извлекает и исполняет команды этой программы без участия центрального процессора и прерывает ЦП только после завершения всей программы ввода/вывода.

2. Рассмотренному в пункте 1 процессору ввода/вывода придается собственная локальная память, при этом возможно управление множеством устройств В/ВЫВ с минимальным привлечением ЦП.

В первом случае МВВ называют *каналом ввода/вывода* (КВВ), а во втором — *процессором ввода/вывода*. В принципе различие между каналом и процессором ввода/вывода достаточно условно, поэтому в дальнейшем будем пользоваться термином «канал».

Концепция системы ввода/вывода с КВВ характерна для больших универсальных вычислительных машин (мэйнфреймов), где проблема

эффективной организации В/ВЫВ и максимального высвобождения центрального процессора в пользу его основной функции стоит наиболее остро. СВВ с каналами ввода/вывода была предложена и реализована в ВМ семейства IBM 360 и получила дальнейшее развитие в семействах IBM 370 и IBM 390.

В ВМ с каналами ввода/вывода центральный процессор практически не участвует в непосредственном управлении внешними устройствами, делегируя эту задачу специализированному процессору, входящему в состав КВВ. Все функции ЦП сводятся к запуску и остановке операций в КВВ, а также проверке состояния канала и подключенных к нему ВУ. Для этих целей ЦП использует лишь несколько (от 4 до 7) команд ввода/вывода. Например, в IBM 360 таких команд четыре:

- «Начать ввод/вывод»;
- «Остановить ввод/вывод»;
- «Проверить ввод/вывод»;
- «Проверить канал».

КВВ реализует операции В/ВЫВ путем выполнения так называемой *канальной программы*. Канальные программы для каждого ВУ, с которым предполагается обмен информацией, описывают нужную последовательность операций ввода/вывода и хранятся в основной памяти ВМ. Роль команд в канальных программах выполняют *управляющие слова канала* (УСК), структура которых отличается от структуры обычной машинной команды. Типовое УСК содержит:

- *код операции*, определяющий для КВВ и ВУ тип операции: «Записать» (вывод информации из ОП в ВУ), «Прочитать» (ввод информации из ВУ в ОП), «Управление» (перемещение головок НМД, магнитной ленты и т. п.);

- *указатели* — дополнительные предписания, задающие более сложную последовательность операций В/ВЫВ, например при вводе пропускать отдельные записи или наоборот — с помощью одной команды вводить «разбросанный» по ОП массив как единый;

- *адрес данных*, указывающий область памяти, используемую в операции ввода/вывода;

- *счетчик данных*, хранящий значение длины передаваемого блока данных.

Кроме того, в УСК может содержаться идентификатор ВУ и информация о его уровне приоритета, указания по действиям, которые должны быть произведены при возникновении ошибок и т. п.

Центральный процессор инициирует ввод/вывод путем инструктирования канала о необходимости выполнить канальную программу, находящуюся в ОП, и указания начального адреса этой программы в памяти ВМ. КВВ следует этим указаниям и управляет пересылкой данных. Отметим, что пересылка информации каналом ведется в режиме прямого доступа к памяти. ВУ взаимодействуют с каналом, получая от него приказы. Таким образом, в ВМ с КВВ управление вводом/выводом

строится иерархическим образом. В операциях ввода/вывода участвуют три типа устройств:

- процессор (первый уровень управления);
- канал ввода/вывода (второй уровень);
- внешнее устройство (третий уровень).

Каждому типу устройств соответствует свой вид управляющей информации:

- процессору — команды ввода/вывода;
- каналу - управляющие слова канала;
- периферийному устройству - приказы.

Структура ВМ с канальной системой ввода/вывода показана на рис. 14.5.1.

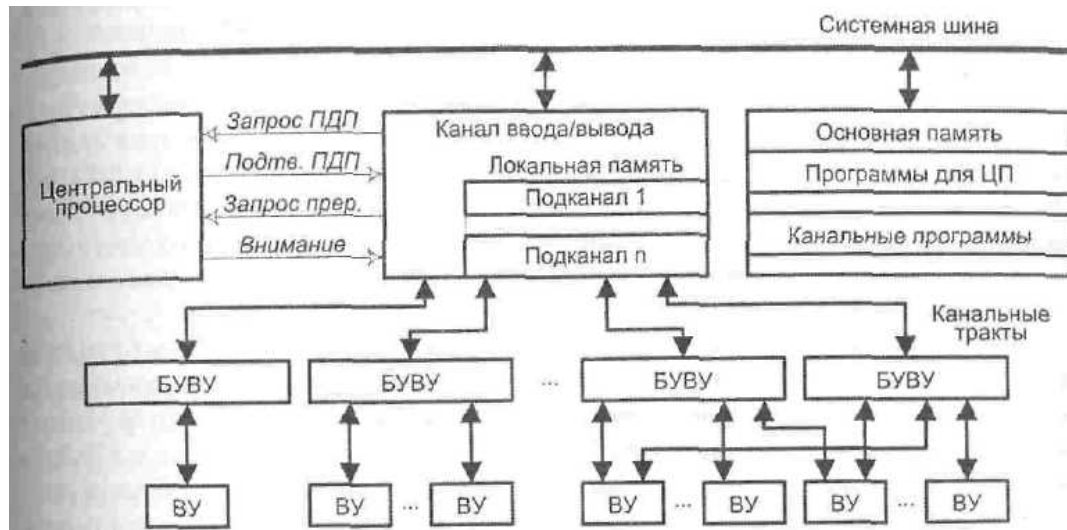


Рис. 14.5.1. ВМ с канальной системой ввода/вывода

Обмен информацией между КВВ и основной памятью осуществляется посредством системной шины ВМ. ВУ подключаются к каналу не непосредственно, а через блоки управления внешними устройствами (БУВУ). БУВУ принимает от канала приказы по управлению внешним устройством (чтение, запись, перемещение носителя или магнитной головки и т.п.) и преобразует их в сигналы управления, свойственные данному типу ВУ. Обычно один БУВУ может обслуживать несколько однотипных ВУ, но для подключения быстродействующих внешних устройств Часто применяются индивидуальные блоки, управления. В свою очередь, некоторые ВУ могут подключаться одновременно к нескольким БУВУ, Это позволяет воспользоваться свободным трактом другого БУВУ при занятости данного БУВУ обслуживанием одного из подключенных к нему ПУ. Физически БУВУ может быть самостоятельным устройством или интегрирован с ВУ или каналом.

Обмен информацией между БУВУ и КВВ обеспечивается так называемыми *канальными трактами*. Обычно каждое БУВУ связано с одним из канальных трактов, но возможно также подключение блока управления сразу к нескольким тракам, что дает возможность избежать нежелательных задержек при занятости одного из них.

В пределах канала ввода/вывода считается, что каждое ВУ подключено к своему *подканалу*. Подканалы имеют свои уникальные логические номера, с помощью которых канальная программа адресуется к конкретному ВУ. Физически подканал реализуется в виде участка памяти, в котором хранятся параметры операции ввода/вывода, выполняемой данным ВУ: текущие значения адреса и счетчика данных, код и указатели операции ввода/вывода, адрес следующего УСК и др. Для хранения этих параметров обычно используется локальная память канала.

Обмен информацией между ВУ и ОП, как уже упоминалось, реализуется в режиме прямого доступа к памяти, при этом для взаимодействия ЦП и канала задействованы сигналы «Запрос ПДП» и «Подтверждение ПДП».

Чтобы известить ЦП об окончании текущей канальной программы или об ошибках, возникших при ее выполнении, КВВ выдает в ЦП сигнал «Запрос прерывания». В свою очередь, ЦП может привлечь внимание канала сигналом «Внимание».

Способ организации взаимодействия ВУ с каналом определяется соотношением быстродействия ОП и ВУ. По этому признаку ВУ образуют две группы: быстродействующие (накопители на магнитных дисках (НМД), накопители на магнитных лентах (НМЛ)) со скоростью приема и выдачи информации около 1 Мбайт/с и медленнодействующие (дисплеи, печатающие устройства и др.) со скоростями порядка 1 Кбайт/с и менее. Быстродействие основной памяти обычно значительно выше. С учетом производительности ВУ в КВВ реализуются два режима *работы*: *мультиплексный* (режим разделения времени) и *монопольный*.

В *мультиплексном режиме* несколько внешних устройств разделяют канал во времени, при этом каждое из параллельно работающих с каналом ВУ связывается с КВВ на короткие промежутки времени только после того, как ВУ будет готово к приему или выдаче очередной порции информации (байта, группы байтов и т. д.). Такая схема принята в *мультиплексном канале ввода/вывода*. Если в течение сеанса связи пересылается один байт или несколько байтов, образующих одно машинное слово, канал называется *байт-мультиплексным*. Канал, в котором в пределах сеанса связи пересылка данных выполняется поблочно, носит название *блок-4 мультиплексного*.

В *монопольном режиме* после установления связи между каналом и ВУ последнее монополизировало канал на все время до завершения инициированной процессором канальной программы и всех предусмотренных этой программой пересылок данных между ВУ и ОП. На все время выполнения канальной программы канал оказывается недоступным для других ВУ. Данную процедуру обеспечивает *селекторный канал ввода/вывода*. Отметим, что в блок-мультиплексном канале в рамках сеанса связи пересылка блока осуществляется в монопольном режиме.

14.6 Канальная подсистема

В последовавших за IBM 360 семействах универсальных ВМ семейства IBM 370 и особенно IBM 390 концепция системы ввода/вывода на базе каналов получила дальнейшее развитие и вылилась в так называемую *канальную подсистему ввода/вывода* (КПВВ). Главная идея заключается в интегрировании отдельных КВВ в единый специализированный процессор ввода/вывода с большим числом канальных трактов и подканалов. Блоки управления внешними устройствами обычно подключаются к нескольким канальным трактам, что позволяет динамически менять путь пересылки информации с учётом текущей их загруженности. Кроме того, разные канальные тракты могут обладать различной пропускной способностью при выборе трактов для подключения определенных ВУ может быть учтено и быстроедействие.

Одной из наиболее совершенных канальных подсистем обладают ВМ семейства IBM 390. В ней предусмотрено использование до 65 536 подканалов и до 256 канальных трактов. Реализованы два типа канальных трактов: параллельный и последовательный.

Параллельные канальные тракты по своим возможностям и принципу действия аналогичны рассмотренным ранее мультиплексному и селекторному каналам, но в отличие от них являются универсальными, то есть могут работать в байт-мультиплексном, блок-мультиплексном и селекторном режимах. Такие канальные тракты в КПВВ называют параллельными, поскольку они обеспечивают пересылку информации параллельным кодом.

Для работы с ВУ, соединенными с КПВВ волоконно-оптическими линиями, используются *последовательные канальные тракты*, реализующие протокол ESCON (Enterprise Systems Connection Architecture). Последовательный канальный тракт рассчитан на передачу информации только в последовательном коде и только в селекторном режиме. Для подключения блоков управления внешними устройствами (БУВУ) к ESCON-тракту служат специальные устройства, называемые ESCON-директорами. Каждое такое устройство может обеспечить одновременное подключение до 60 БУВУ и одновременную передачу информации от 30 из них со скоростью до 10 Мбайт/с.

Кроме того, в КПВВ предусмотрены специальные коммуникационные канальные тракты для подключения к сетям ВМ, модемам, другим системам.

В принципе основное преимущество КПВВ - динамическое перераспределение канальных трактов - в какой-то мере может быть реализовано и в рамках каждого отдельного канала. Однако объединение всех канальных ресурсов в единую канальную подсистему позволяет применить оптимальную стратегию динамического распределения и использования этих ресурсов и благодаря этому достичь качественно нового уровня эффективности системы ввода/вывода.