

Лекция 5. Системы ввода-вывода

Необходимым элементом архитектуры микропроцессорной системы является система ввода/вывода (СВВ). Система ввода/вывода призвана обеспечить обмен информацией между ядром микропроцессорной системы и разнообразными внешними устройствами (ВУ). Технические и программные средства СВВ несут ответственность за физическое и логическое сопряжение ядра вычислительной машины и внешних устройств.

Системам ввода/вывода по сравнению с прочими элементами архитектуры вычислительных машин уделялось несколько меньшее внимание. Подтверждением этого можно считать то, что многие программы контроля производительности не учитывают влияние операций ввода/вывода на эффективность микропроцессорной системы. Вследствие этого существует разрыв в производительности процессора и скоростью ввода/вывода.

Системы ввода/вывода

Технически система ввода/вывода в рамках микропроцессорной системы реализуется комплексом *модулей ввода/вывода* (МВВ).

Модуль ввода/вывода выполняет сопряжение ВУ с ядром микропроцессорной системы и различные коммуникационные операции между ними.

Две основные функции МВВ:

- обеспечение интерфейса с ЦП и памятью (*«большой» интерфейс*);
- обеспечение интерфейса с одним или несколькими периферийными устройствами (*«малый» интерфейс*).

Способы подключения СВВ к процессору

1 способ. Раздельные шины памяти ввода-вывода



В варианте с отдельными шинами памяти и ввода/вывода обмен информацией между ЦП и памятью физически отделен от ввода/вывода, поскольку обеспечивается полностью независимыми шинами.

Достоинства:

- возможность осуществлять обращение к памяти одновременно с выполнением ввода/вывода.
- данный архитектурный вариант ВМ позволяет специализировать каждую из шин,
- учесть формат пересылаемых данных,
- особенности синхронизации обмена и т. п.
- в частности, шина ввода/вывода, с учетом характеристик реальных ЗУ, может иметь меньшую пропускную способность, что позволяет снизить затраты на ее реализацию.

Недостатком способа можно считать большое количество точек подключения к ЦП.

2 способ. Подключение СВВ с совместно используемыми линиями данных и адреса

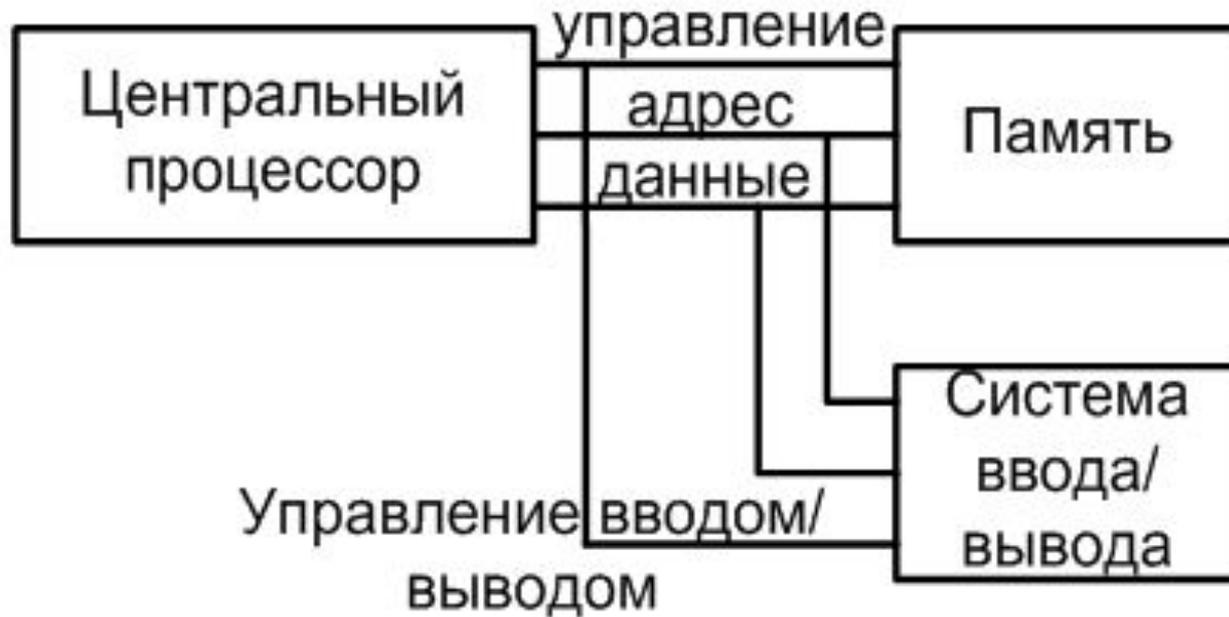


Достоинства недостатки 2 способа

Память и СВВ имеют общие для них линии адреса и линии данных разделяя их во времени. В то же время управление памятью и СВВ, а также синхронизация их взаимодействия с процессором осуществляются независимо по отдельным линиям управления.

Это позволяет учесть особенности процедур обращения к памяти и к модулям ввода/вывода и добиться наибольшей эффективности доступа к ячейкам памяти и внешним устройствам.

3 способ. Подключение СВВ подключение на общих правах с процессором и памятью



Последний тип архитектуры VM предполагает подключение СВВ к системной шине на общих правах с процессором и памятью. Для такого подхода характерны простота и низкая стоимость, однако такая организация не в состоянии обеспечить высокие интенсивность и скорость транзакций.

Адресное пространство системы ввода-вывода

Как и обращение к памяти, операции ввода/вывода также предполагают наличие некоторой системы адресации, позволяющей выбрать один из модулей СВВ, а также одно из подключенных к нему внешних устройств. Адрес модуля и ВУ является составной частью соответствующей команды, в то время как расположение данных на внешнем устройстве определяется пересылаемой на ВУ информацией.

Адресное пространство ввода/вывода может быть совмещено с адресным пространством памяти или быть выделенным. При совмещении адресного пространства для адресации модулей ввода/вывода отводится определенная область адресов

Распределение совмещенного адресного пространства



Совмещенное адресное пространство используется в вычислительных машинах MIPS и SPARC.

Достоинства совмещенного адресного пространства:

- расширение набора команд для обращения к внешним устройствам, позволяет сократить длину программы и повысить быстродействие;
- значительное увеличение количества подключаемых внешних устройств;
- возможность внепроцессорного обмена данными между внешними устройствами, если в системе команд есть команды пересылки между ячейками памяти;
- возможность обмена информацией не только с аккумулятором, но и с любым регистром центрального процессора.

Недостатки совмещенного адресного пространства:

- сокращение области адресного пространства памяти;
- усложнение декодирующих схем адресов в СВВ;
- трудности распознавания операций передачи информации при вводе/выводе среди других операций. Сложности в чтении и отладке программы, в которой простые команды вызывают выполнение сложных операций ввода/вывода;
- трудности при построении СВВ на простых модулях ввода/вывода: сигналы управления не смогут координировать сложную процедуру ввода/вывода. Поэтому МВВ часто должны генерировать дополнительные сигналы под управлением программы.

Достоинства выделенного адресного пространства:

- адрес внешнего устройства в команде ввода/вывода может быть коротким. В большинстве СВВ количество внешних устройств намного меньше количества ячеек памяти. Короткий адрес ВУ подразумевает такие же короткие команды ввода/вывода и простые дешифраторы;
- программы становятся более наглядными, так как операции ввода/вывода выполняются с помощью специальных команд;
- разработка СВВ может проводиться отдельно от разработки памяти.

Недостатки выделенного адресного пространства:

- ввод/вывод производится только через аккумулятор центрального процессора. Для передачи информации от ВУ в РОН, если аккумулятор занят, требуется выполнение четырех команд (сохранение содержимого аккумулятора, ввод из ВУ, пересылка из аккумулятора в РОН, восстановление содержимого аккумулятора);
- перед обработкой содержимого ВУ это содержимое нужно переслать в ЦП.

Подключение внешних устройств

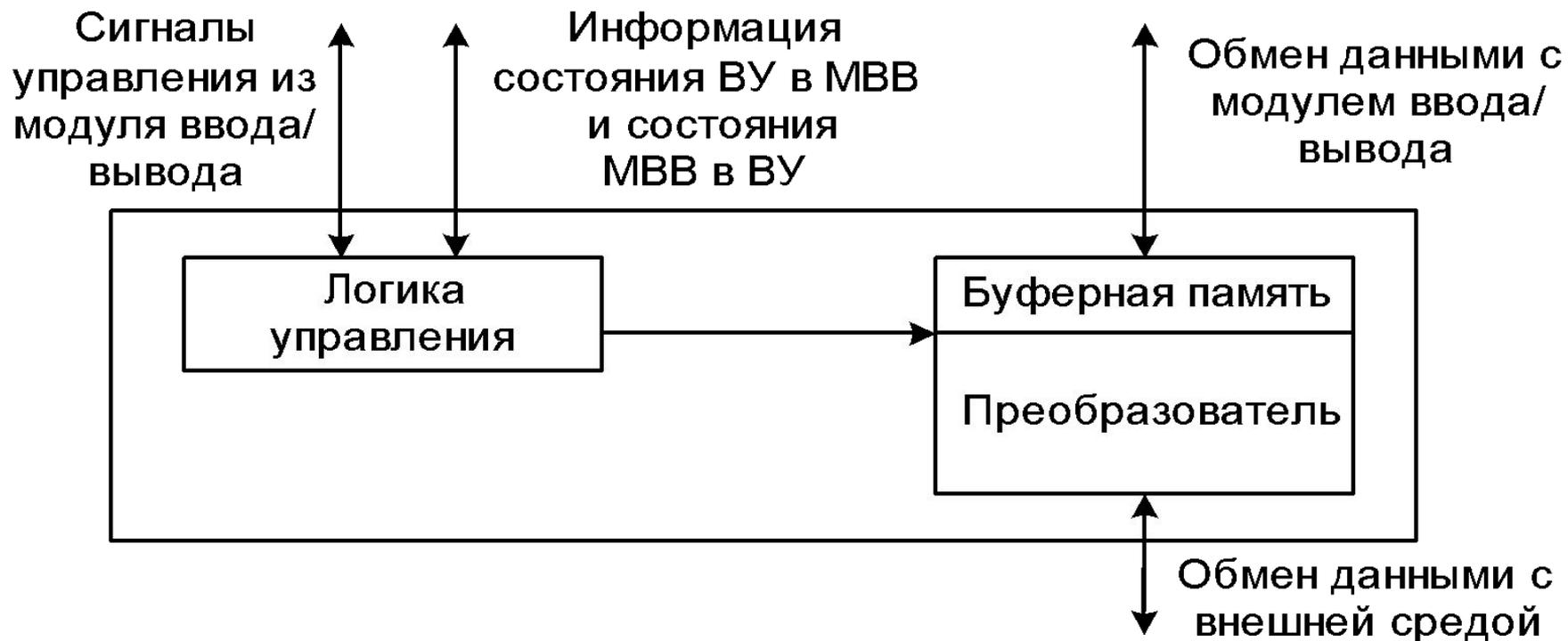
Связь ВМ с внешним миром осуществляется с помощью самых разнообразных внешних устройств. Каждое ВУ подключается к МВВ посредством индивидуальной шины. Интерфейс, по которому организуется такое взаимодействие МВВ и ВУ, часто называют *малым*. Индивидуальная шина обеспечивает обмен данными и управляющими сигналами, а также информацией о состоянии участников обмена.

Внешнее устройство, подключенное к МВВ, обычно называют *периферийным устройством* (ПУ).

Структура внешнего устройства

Все множество ПУ можно свести к трем категориям:

- для общения с пользователем;
- для общения с ВМ;
- для связи с удаленными устройствами.



Интерфейс с МВВ реализуется в виде сигналов управления, состояния и данных. Данные представлены совокупностью битов, которые должны быть переданы в модуль ввода/вывода или получены из него. *Сигналы управления* определяют функцию, которая должна быть выполнена внешним устройством. Это может быть стандартная для всех устройств функция — посылка данных в МВВ или получение данных из него, либо специфичная для данного типа ВУ функция, например, позиционирование считывателя магнитного диска или перемотка магнитной ленты. *Сигналы состояния* характеризуют текущее состояние устройства, в частности включено ли ВУ и готово ли оно к передаче данных.

Логика управления — это схемы, координирующие работу ВУ в соответствии с направлением передачи данных. Задачей *преобразователя* является трансформация информационных сигналов, имеющих самую различную физическую природу, в электрические сигналы, а также обратное преобразование.

Обычно совместно с преобразователем используется *буферная память*, обеспечивающая временное хранение данных, пересылаемых между МВВ и ВУ.

Модули ввода/вывода. Функции модуля

Модуль ввода/вывода в составе МПС отвечает за управление одним или несколькими ВУ и за обмен данными между этими устройствами с одной стороны, и основной памятью или регистрами ЦП — с другой. Основные функции МВВ можно сформулировать следующим образом:

- локализация данных;
- управление и синхронизация;
- обмен информацией;
- буферизация данных;
- обнаружение ошибок.

Под **локализацией данных** понимают возможность обращения к одному из ВУ, а также адресации данных на нем.

Адрес ВУ обычно содержится в адресной части команд ввода/вывода. В состав СВВ могут входить несколько модулей ввода/вывода. Каждому модулю назначается определенный диапазон адресов, независимо от того, является ли пространство адресов совмещенным или отдельным. Старшие разряды в адресах диапазона, выделенного модулю, обычно одинаковы и обеспечивают выбор одного из МВВ в рамках системы ввода/вывода. Младшие разряды адреса представляют собой уникальные адреса регистров данного модуля или подключенных к нему ВУ. Одной из функций МВВ является проверка вхождения поступившего по шине адреса в выделенный данному модулю диапазон адресов. При положительном ответе модуль должен обеспечить дешифровку поступившего адреса и перенаправление информации к адресуемому объекту или от него.

Для простейших внешних устройств (клавиатура, принтер и т. п.) адрес ВУ однозначно определяет и расположение данных на этом устройстве. Для более сложных ВУ, таких как внешние запоминающие устройства, информация о местонахождении данных требует детализации. Так, для ЗУ на магнитной ленте необходимо указать номер записи, а для магнитного диска — номер цилиндра, номер сектора и т. п. Эта часть адресной информации передается в МВВ не по шине адреса, а в виде служебных сообщений, пересылаемых по шине данных. Обработка такой информации в модуле, естественно, сложнее, чем выбор нужного регистра или ВУ. В частности, она может требовать от МВВ организации процедуры поиска на носителе информации.

Управление и синхронизация

Функция управления и синхронизации заключается в том, что МВВ должен координировать перемещение данных между внутренними ресурсами ВМ и внешними устройствами. При разработке системы управления и синхронизации модуля вв. да/вывода необходимо учитывать целый ряд факторов.

Прежде всего, нужно принимать во внимание, что ЦП может взаимодействовать одновременно с несколькими ВУ, причем быстродействие подключаемых к МВВ внешних устройств варьируется в очень широких пределах — от нескольких байтов в секунду в терминалах до десятков миллионов байтов в секунду при обмене с магнитными дисками. Если в системе используются шины, каждое взаимодействие между ЦП и МВВ включает в себя одну или несколько процедур арбитража.

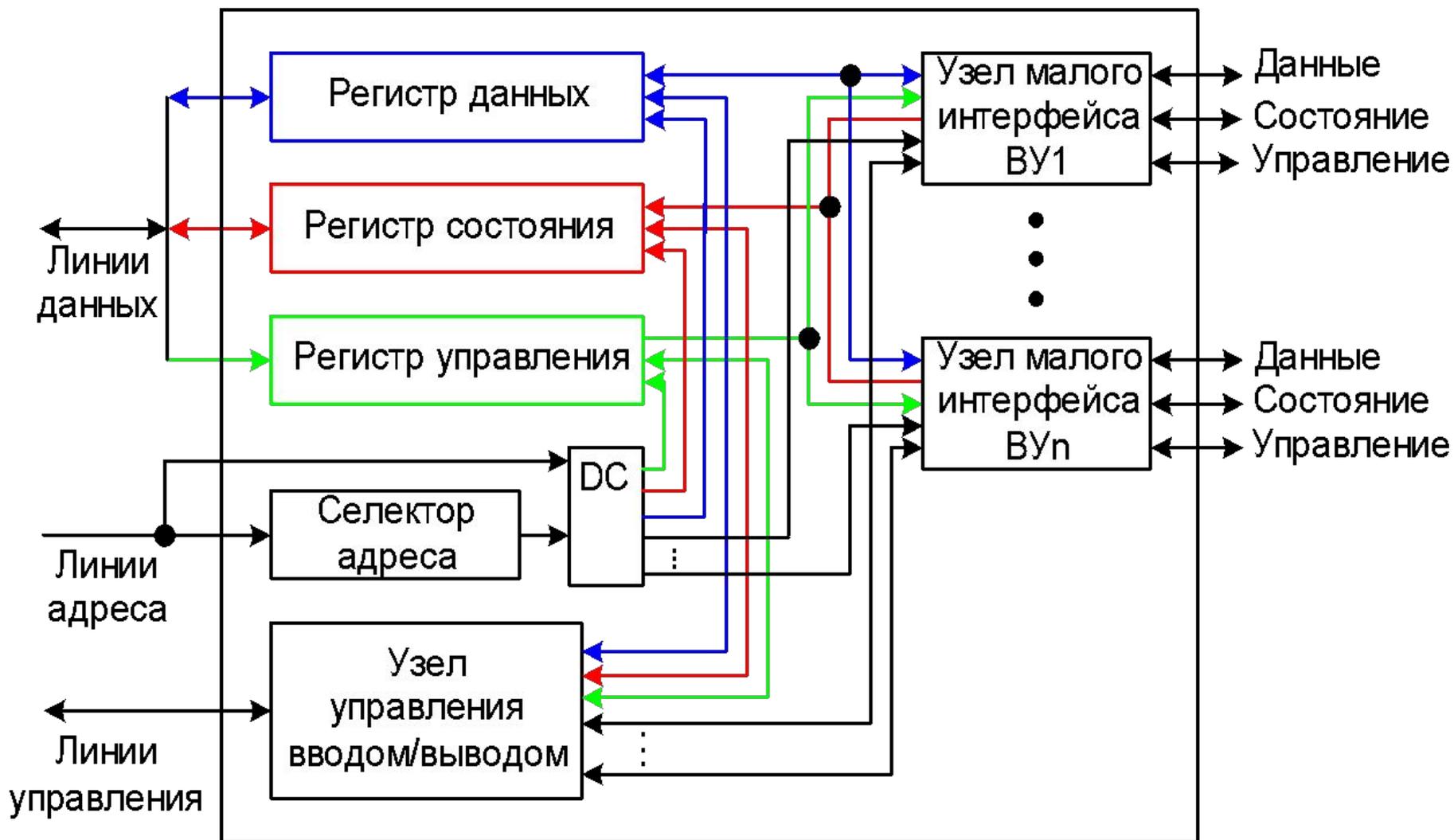
- В отличие от обмена с памятью процессы ввода/вывода и работа ЦП протекают несинхронно. Очередная порция информация может быть выдана на устройство вывода лишь тогда, когда это устройство готово их принять. Аналогично, ввод от устройства ввода допустим только в случае доступности информации на устройстве ввода.

Таким образом, модуль ввода/вывода обязан снабдить центральный процессор информацией о собственной готовности к обмену, а также о готовности подключенных к модулю ВУ. Помимо этого, процессор должен обладать оперативными сведениями и об иных происходящих в СВВ событиях.

Структура модуля ввода/вывода

«Большой» интерфейс

«Малый» интерфейс



Структура MBV в значительной мере зависит от числа и сложности внешних устройств, которыми он управляет, однако в самом общем виде такой модуль можно представить в форме, показанной на рис. 8.5.

Связь модуля ввода/вывода с ядром VM осуществляется посредством системной или специализированной шины. С этой стороны в MBV реализуется так называемый «большой» интерфейс. Большие различия в системе команд и шин VM являются причиной того, что со стороны «большого» интерфейса модули ввода/вывода трудно унифицировать, и часто MBV, созданные для одних VM, не могут быть использованы в других. Тем не менее, по структуре они схожи. Данные, передаваемые в модуль и из него, буферизируются в регистре данных. Буферизация позволяет компенсировать различие в быстродействии ядра VM и внешних устройств. Разрядность регистра, как правило, совпадает с шириной шины данных со стороны «большого» интерфейса (2, 4 или 8 байт).

В свою очередь, большинство ВУ ориентировано на побайтовый обмен информацией. Побайтовая пересылка информации по «широкой» системной шине — крайне неэффективное решение, поэтому со стороны «малого» интерфейса регистр данных часто дополняют узлом упаковки/распаковки (на схеме не показан). Этот узел при вводе обеспечивает последовательное побайтовое заполнение регистра данных (упаковку), а при выводе — последовательную побайтовую выдачу содержимого регистра на ВУ (распаковку). В результате при обмене данными через «большой» интерфейс занята вся ширина шины данных. В МВВ, рассчитанных на работу с большим числом ВУ, могут входить несколько регистров данных, что позволяет независимо хранить текущие данные каждого из внешних устройств.

Помимо регистра данных в составе МВВ имеются также регистр управления и регистр состояния (либо совмещенный регистр управления/состояния).

В *регистре управления* (РУ) фиксируются поступившие из ЦП команды управления модулем или подключенными к нему внешними устройствами. Отдельные разряды регистра могут представлять такие команды, как очистка регистров МВВ, сброс ВУ, начало чтения, начало записи и т. п. В сложных МВВ присутствует несколько регистров управления, например регистр управляющих сигналов для модуля в целом и отдельные РУ для каждого из внешних устройств.

Регистр состояния (РС) служит для хранения битов состояния МВВ и подключенных к нему внешних устройств. Содержимое определенного разряда регистра может характеризовать, например, готовность устройства ввода к приему очередной порции данных, занятость устройства вывода или нахождение ВУ в автономном режиме. Не исключается наличие более одного регистра состояния.

Процедура ввода/вывода предполагает возможность работы с каждым регистром МВВ или внешним устройством по отдельности. Такая возможность обеспечивается системой адресации. Каждому модулю в адресном пространстве ввода/вывода выделяется уникальный набор адресов, количество адресов в котором зависит от числа адресуемых элементов. Поступивший из ЦП адрес с помощью селектора адреса проверяется на принадлежность к диапазону, выделенному данному МВВ. В случае подтверждения дешифратор DC выполняет раскодирование адреса, разрешая работу с соответствующим регистром модуля или ВУ.

Узел управления вводом/выводом по сути играет роль местного устройства управления МВВ. На него возлагаются две задачи: обеспечение взаимодействия с ЦП и координация работы всех составляющих МВВ. Связь с ЦП реализуется посредством линий управления, по которым из ЦП в модуль поступают сигналы, служащие для синхронизации операций ввода и вывода. В обратном направлении передаются сигналы, информирующие о происходящих в модуле событиях, например, сигналы прерывания. Часть линий управления может задействоваться модулем для арбитража. Вторая функция узла управления реализуется с помощью внутренних сигналов управления.

Со стороны «малого» интерфейса МВВ обеспечивает подключение внешних устройств и взаимодействие с ними. Эта часть МВВ более унифицирована, поскольку внешние устройства всегда подгоняются под один из стандартных протоколов. Каждое из внешних устройств «обслуживается» своим узлом «малого» интерфейса, который реализует принятый для данного ВУ стандартный протокол взаимодействия. При управлении широким спектром ВУ модуль должен по возможности освобождать ЦП от знания деталей конкретных ВУ, так чтобы ЦП мог управлять любым устройством с помощью команд чтения и записи. МВВ при этом берет на себя задачи синхронизации, согласования форматов данных и т. п.

Модуль ввода/вывода, который берет на себя детальное управление ВУ и общается с ЦП только с помощью команд высокого уровня, часто называют *каналом ввода/вывода* или *процессором ввода/вывода*. Простой МВВ, требующий детального управления со стороны ЦП, называют *контроллером ввода/вывода* или *контроллером устройства*. Как правило, контроллеры ввода/вывода типичны для микро-ЭВМ, а каналы ввода/вывода — для универсальных ВМ.

Методы управления вводом/выводом

В ВМ находят применение три способа организации ввода/вывода:

- 1) программно управляемый ввод/вывод;
- 2) ввод/вывод по прерываниям;
- 3) прямой доступ к памяти.

При **программно управляемом вводе/выводе** все 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, ЦП 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 не 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, извещающий о завершении выполнения ранее выданной команды. Как и при программно управляемом вводе/выводе, ЦП отвечает за извлечение данных из памяти (при выводе) и запись данных в память (при вводе).

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

Программно управляемый ввод/вывод

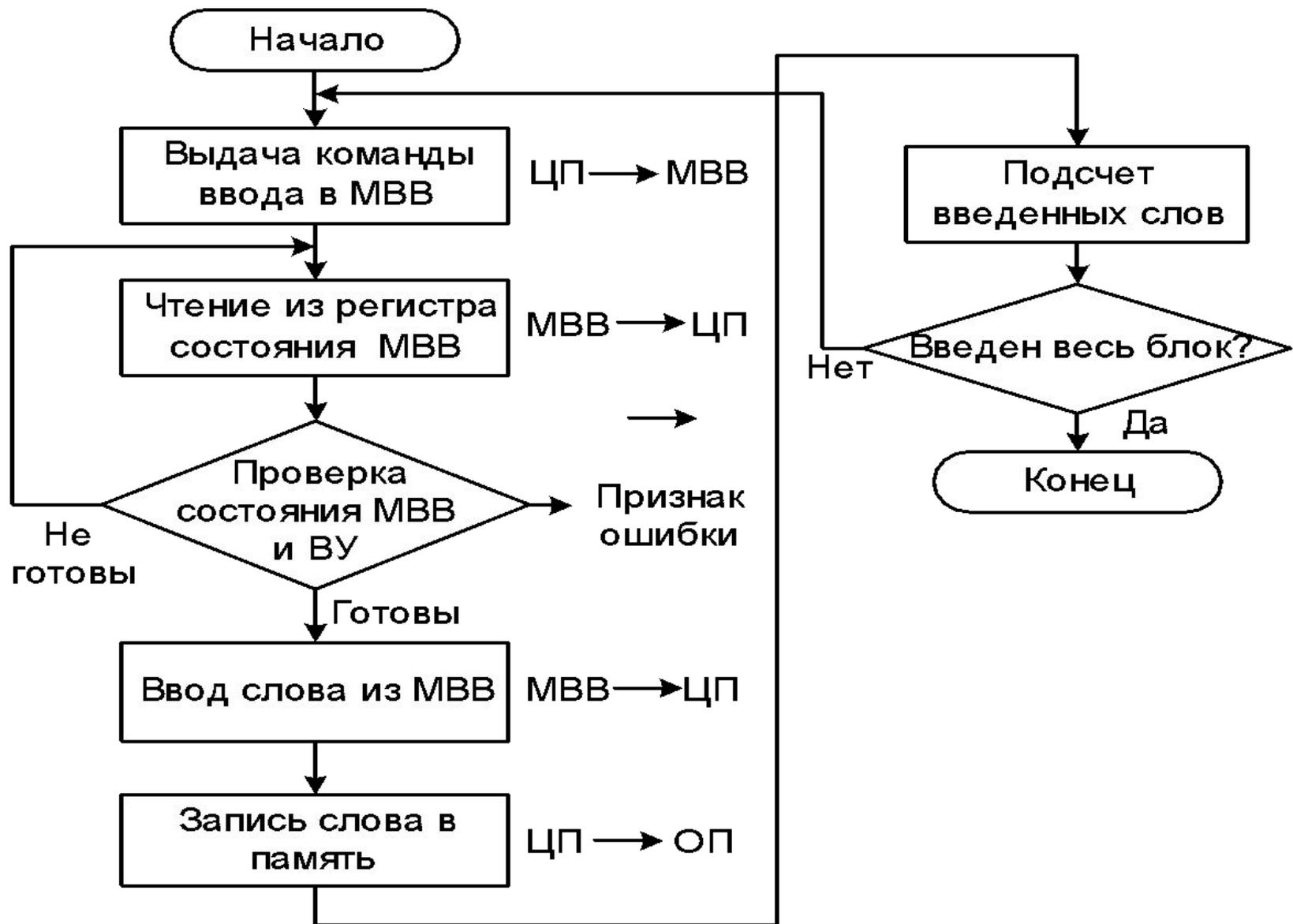
Наиболее простым методом управления вводом/выводом является *программно управляемый ввод/вывод*, часто называемый также *вводом/выводом с опросом*.

Ввод/вывод происходит:

- 1) под полным контролем центрального процессора,
- 2) 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. Для определения момента завершения операции или пересылки очередного элемента блока данных процессор должен периодически опрашивать и анализировать содержимое регистра состояния МВВ.

Программно-управляемый ввод данных



Существуют четыре типа команд В/ВЫВ, которые может получить МВВ: управление, проверка, чтение и запись.

- 1) **Команды управления** используются для активизации ВУ и указания требуемой операции. Например, в устройство памяти на магнитной ленте может быть выдана команда перемотки или продвижения на одну запись. Для каждого типа ВУ характерны специфичные для него команды управления.
- 2) **Команда проверки** применяется для проверки различных ситуаций, возникающих в МВБ и ВУ в процессе ввода/вывода. С помощью таких команд ЦП способен выяснить, включено ли ВУ, готово ли оно к работе, завершена ли последняя операция ввода/вывода и не возникли ли в ходе ее выполнения какие-либо ошибки. Действие команды сводится к установке или сбросу соответствующих разрядов регистра состояния МВВ.
- 3) **Команда чтения** побуждает модуль получить элемент данных из ВУ и занести его в регистр данных (РД). ЦП может получить этот элемент данных, запросив МВВ поместить его на шину данных.
- 4) **Команда записи** заставляет модуль принять элемент данных (байт или слово) с шины данных и переслать его в РД с последующей передачей в ВУ.

Из блок-схемы виден основной недостаток программно управляемого ввода/вывода — 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 МВВ, запись в МВВ битов управления, чтение или запись данных со скоростью, определяемой внешним устройством. Все это также отрицательно сказывается на эффективности ввода/вывода.

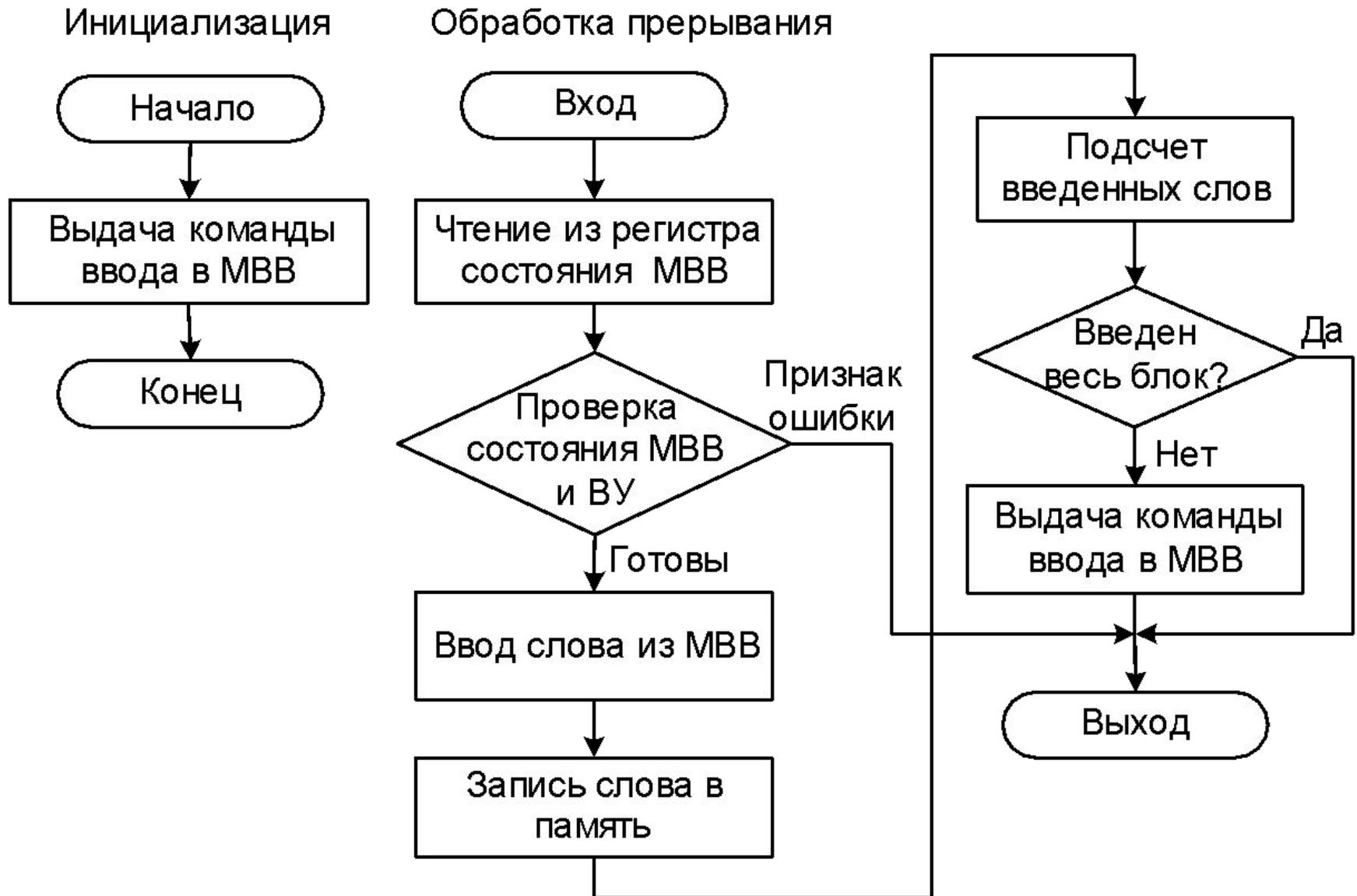
Главным аргументом в пользу программно управляемого ввода/вывода является простота МВВ, поскольку основные функции по управлению вводом/выводом берет на себя процессор. При одновременной работе с несколькими ВУ приоритет устройств легко изменить программными средствами (последовательностью опроса).

Ввод/вывод по прерываниям

Альтернативой программного ввода-вывода может быть вариант, когда ЦП выдает команду ввода/вывода, а затем продолжает делать другую полезную работу. Когда ВУ готово к обмену данными, оно через МВВ извещает об этом процессор с помощью запроса на прерывание. ЦП осуществляет передачу очередного элемента данных, после чего возобновляет выполнение прерванной программы.

Схема алгоритма ввода блока данных с использованием ввода/вывода по прерываниям приведена на рисунке 8.7. Оставим без внимания такие подробности, как сохранение и восстановления контекста, действия, выполняемые при завершении пересылки блока данных, а также в случае возникновения ошибок.

Ввод данных по прерыванию



Процедура ввода блока данных по прерываниям реализуется следующим образом. ЦП выдает команду чтения, а затем продолжает выполнение других заданий, например другой программы. Получив команду, МВВ приступает к вводу элемента данных с ВУ. Когда считанное слово оказывается в регистре данных модуля, МВВ формирует на управляющей линии сигнал прерывания ЦП. Выставив запрос, МВВ помещает введенную информацию на шину данных, после чего он готов к следующей операции ввода/вывода.

ЦП в конце каждого цикла команды проверяет наличие запросов прерывания. Когда от МВВ приходит такой сигнал, ЦП сохраняет контекст текущей программы и обрабатывает прерывание. В рассматриваемом случае ЦП читает слово из модуля, записывает его в память и выдает модулю команду на считывание очередного слова. Далее ЦП восстанавливает контекст прерванной программы и возобновляет ее выполнение.

Реализация ввода/вывода по прерываниям

При реализации ввода/вывода по прерываниям необходимо дать ответы на 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.

Рассмотрим вопрос идентификации устройства. Здесь возможны три основных метода:

- 1) множественные линии прерывания;
- 2) программная идентификация;
- 3) векторное прерывание.

Наиболее **простой подход** к решению проблемы определения источника запроса — применение **множественных линий прерывания** между ЦП и модулями ввода/вывода, хотя выделение слишком большого количества управляющих линий для этих целей нерационально. Если присутствует несколько линий прерывания, желательно, чтобы каждая линия использовалась всеми МВВ, при этом для каждой линии действует один из двух остальных методов идентификации устройства.

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, выставлял он запрос или нет.

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, указатель на соответствующую программу обработки прерывания. Реализуется он с помощью хранящейся в памяти **таблицы векторов прерывания**, где содержатся адреса программ обработки прерываний.

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;
- в ЦП вовлечен в управление передачей, для каждой пересылки он должен выполнить определенное количество команд.

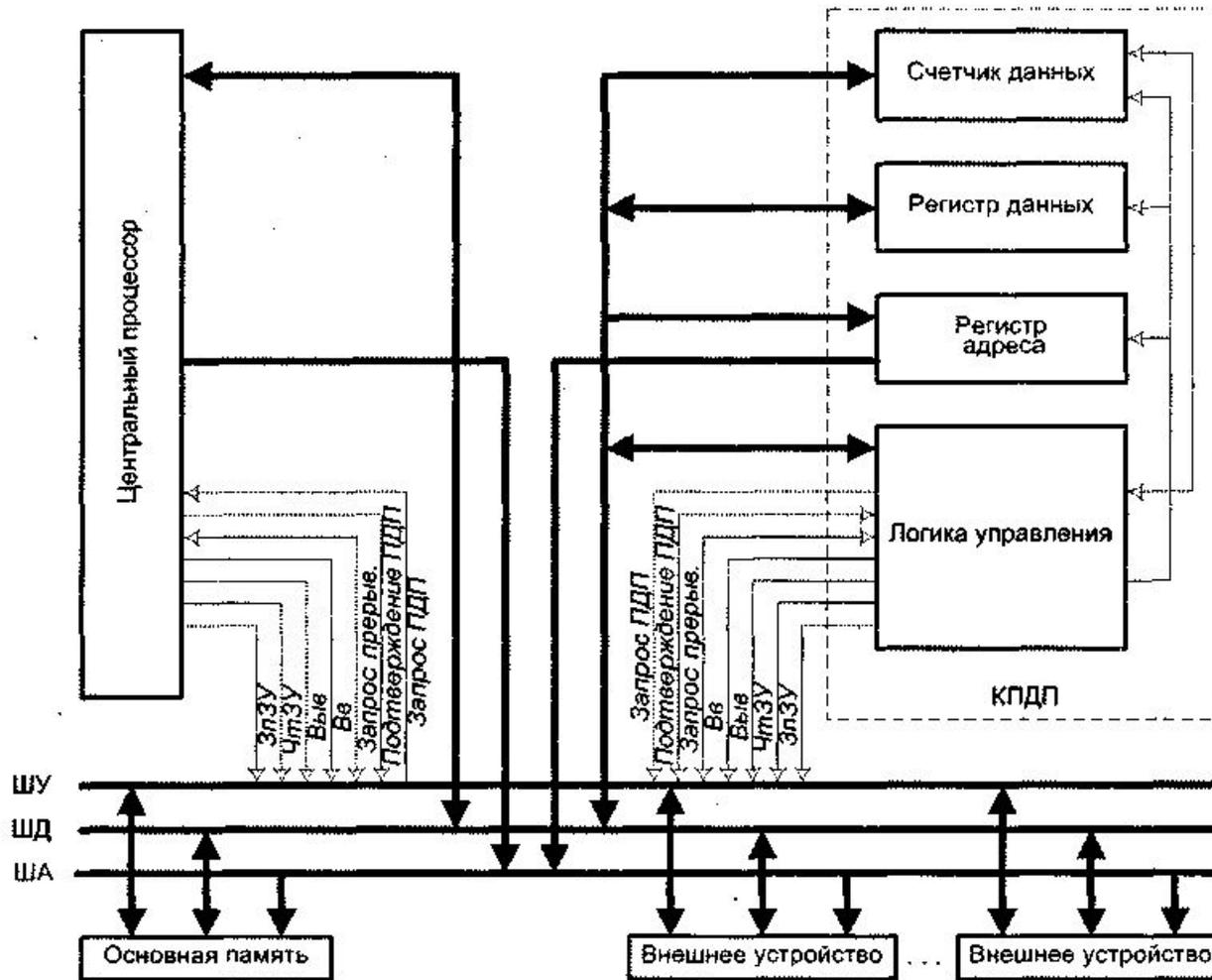
Когда пересылаются большие объемы данных, требуется более эффективный способ ввода/вывода — *прямой доступ к памяти* (ПДП). ПДП предполагает наличие на системной шине дополнительного модуля — *контроллера прямого доступа к памяти* (КПДП), способного брать на себя функции ЦП по управлению системной шиной 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:

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

Первый параметр определяет направление пересылки данных: из ОП в ВУ или наоборот. За исходную точку обычно принимается память, поэтому под чтением понимают считывание данных из ОП и выдачу их на устройство вывода, а под записью — прием данных из устройства ввода и запись в ОП. Вид запроса запоминается в схеме логики управления контроллера.

Организация прямого доступа к памяти



Точки возможного вмешательства в цикл команды при прямом доступе к памяти и при обычном прерывании



Возможные конфигурации систем прямого доступа к памяти

