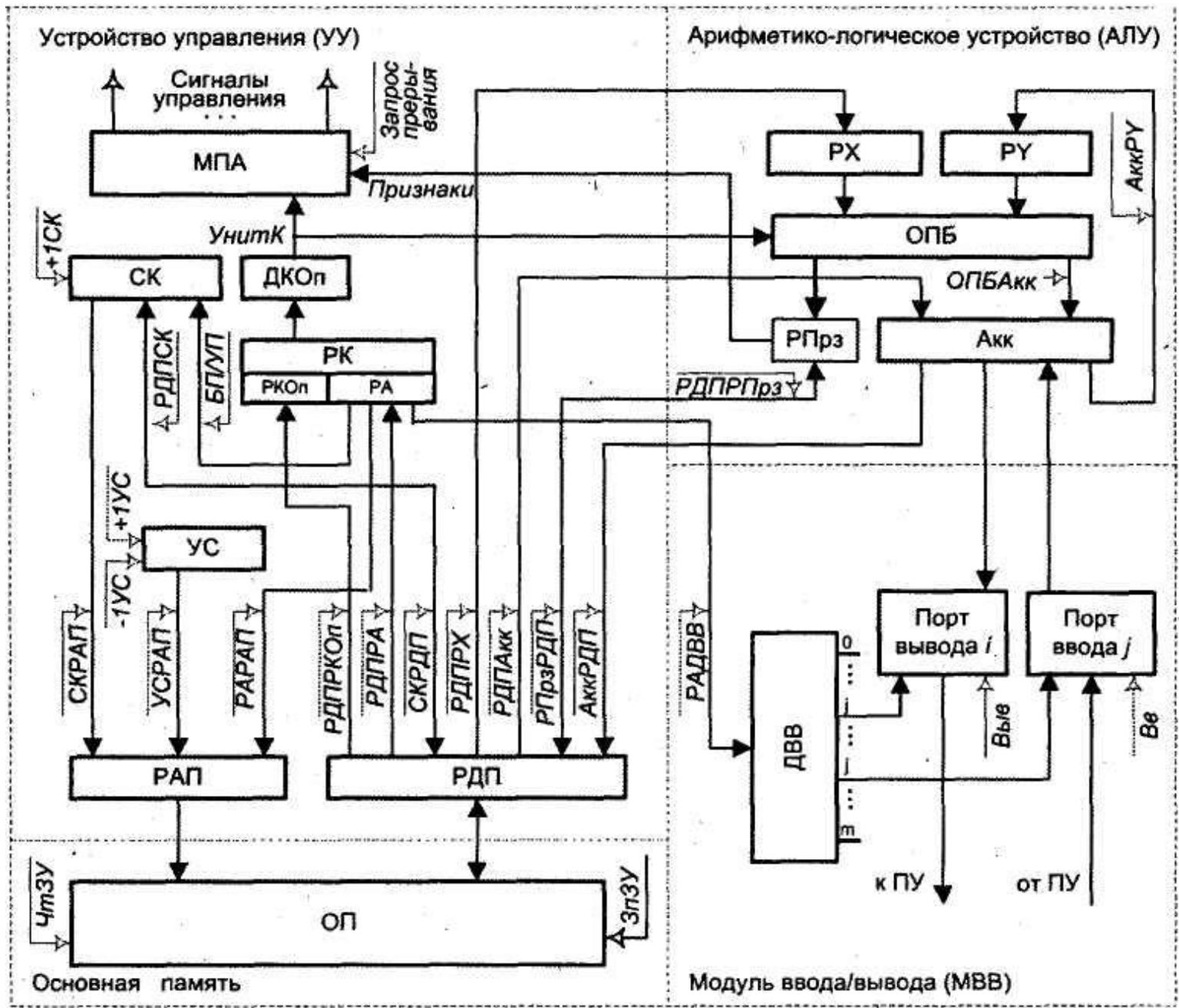


**Функциональная
организация
фон-неймановской ВМ**

Лекция №3



Полный цикл команды может включать в себя следующие состояния:

- **Вычисление адреса команды.** Определение исполнительного адреса команды, которая должна выполняться следующей.
- **Выборка команды.** Чтение команды из ячейки памяти и занесение ее в РК.
- **Декодирование команды.** Анализ команды с целью выяснения типа подлежащей выполнению операции и операндов.
- **Вычисление адреса операнда.** Определение исполнительного адреса операнда, если операция предполагает обращение к операнду, хранящемуся в памяти или же доступному посредством ввода.
- **Выборка операнда.** Выборка операнда из памяти или его ввод с устройства ввода.
- **Операция с данными.** Выполнение операции, указанной в команде.
- **Запись операнда.** Запись результата в память или вывод на устройство вывода.

Счетчик команд (СК) — неотъемлемый элемент устройства управления любой ВМ, построенной в соответствии с фон-неймановским принципом программного управления. Согласно этому принципу соседние команды программы располагаются в ячейках памяти со следующими по порядку адресами и выполняются преимущественно в той же очередности, в какой они размещены в памяти ВМ. Таким образом, адрес очередной команды может быть получен путем увеличения адреса ячейки, из которой была считана текущая команда, на длину выполняемой команды, представленную числом занимаемых ею ячеек. Реализацию такого режима и призван обеспечивать счетчик команд — двоичный счетчик, в котором хранится и модифицируется адрес очередной команды программы. Перед началом вычислений в СК заносится адрес ячейки основной памяти, где хранится команда, которая должна быть выполнена первой. В процессе выполнения каждой команды путем увеличения содержимого СК на длину выполняемой команды в счетчике формируется адрес следующей подлежащей выполнению команды. В рассматриваемой ВМ любая команда занимает одну ячейку, поэтому содержимое СК увеличивается на единицу, что обеспечивается подачей сигнала управления +1СК. По завершении текущей команды адрес следующей команды программы всегда берется из счетчика команд. Для изменения естественного порядка вычислений (перехода в иную точку программы) достаточно занести в СК адрес точки перехода.

Разработчики ВМ используют иные названия, в частности *программный счетчик (РС, Program Counter)* или *указатель команды (IP, Instruction Pointer)*. Последнее определение представляется наиболее удачным, поскольку точнее отражает назначение рассматриваемого узла УУ.

Регистр команды

Счетчик команд определяет лишь местоположение команды в памяти, но не содержит информации о том, что это за команда. Чтобы приступить к выполнению команды, ее необходимо извлечь из памяти и разместить *в регистре команды (РК)*. Этот этап носит название *выборки команды*. Только с момента загрузки команды в РК она становится «видимой» для процессора. В РК команда хранится в течение всего времени ее выполнения. Как уже отмечалось ранее, любая команда содержит два поля: поле кода операции и поле адресной части. Учитывая это обстоятельство, регистр команды иногда рассматривают как совокупность двух регистров — *регистра кода операции (РКОп)* и *регистра адреса (РА)*, в которых хранятся соответствующие составляющие команды.

Если команда занимает несколько последовательных ячеек, то код операции всегда находится в том слове команды, которое извлекается из памяти первым. Это позволяет по коду операции определить, требуются ли считывание из памяти и загрузка в РК остальных слов команды. Собственно выполнение команды начинается только после занесения в РК ее полного кода.

Указатель стека

Указатель стека (УС) — это регистр, где хранится адрес вершины стека. В реальных вычислительных машинах стек реализуется в виде участка основной памяти, обычно расположенного в области наибольших адресов. Заполнение стека происходит в сторону уменьшения адресов, при этом вершина стека — это ячейка, куда была произведена последняя по времени запись. Для хранения адреса такой ячейки и предназначен УС.

При выполнении операции *push* (занесение в стек) содержимое УС с помощью сигнала $-1US$ сначала уменьшается на единицу, после чего используется в качестве адреса, по которому производится запись. Соответствующая ячейка становится новой вершиной стека. Считывание из стека (операция *pop*) происходит из ячейки, на которую указывает текущий адрес в УС, после чего содержимое указателя стека сигналом $+1US$ увеличивается на единицу. Таким образом, вершина стека опускается, а считанное слово считается удаленным из стека. Хотя физически считанное слово и осталось в ячейке памяти, при следующей записи в стек оно будет заменено новой информацией.

Регистр адреса памяти (РАП)

предназначен для хранения адреса ячейки основной памяти вплоть до завершения операции (считывание или запись) с этой ячейкой. Наличие РАП позволяет компенсировать различия в быстродействии ОП и прочих устройств машины.

Регистр данных памяти (РДП)

призван компенсировать разницу в быстродействии запоминающих устройств и устройств, выступающих в роли источников и потребителей хранимой информации. В РДП при чтении заносится содержимое ячейки ОП, а при записи — помещается информация, подлежащая сохранению в ячейке ОП. Собственно момент считывания и записи в ячейку определяется сигналами ЧтЗУ и ЗпЗУ соответственно.

Дешифратор кода операции (ДКОп)

преобразует код операции в форму, требуемую для работы микропрограммного автомата (МПА). Информация после декодирования определяет последующие действия МПА, а ее вид зависит от организации МПА. В рассматриваемой ВМ — это унитарный код УнитК.

Часто код операции преобразуется в адрес первой команды микропрограммы, реализующей указанную в команде операцию. С этих позиций ДКОп правильнее было бы назвать не дешифратором, а ***преобразователем кодов.***

Микропрограммный автомат (МПА)

правомочно считать центральным узлом устройства управления. Именно МПА формирует последовательность сигналов управления, в соответствии с которыми производятся все действия, необходимые для выборки из памяти и выполнения команд. Исходной информацией для МПА служат: декодированный код операции, состояние признаков (флагов), характеризующих результат предшествующих вычислений, а также внешние запросы на прерывание текущей программы и переход на программу обслуживания прерывания.

Арифметико-логическое устройство

Это устройство, как следует из его названия, предназначено для арифметической и логической обработки данных. В машине, изображенной на рис. 3.1, оно содержит следующие узлы.

Операционный блок (ОПБ)

представляет собой ту часть АЛУ, которая, собственно, и выполняет арифметические и логические операции над поданными на вход операндами. Выбор конкретной операции из возможного списка операций для данного ОПБ определяется кодом операции команды. В нашей ВМ код операции поступает непосредственно из регистра команды. В реальных машинах КОп зачастую преобразуется в МПА в иную форму и уже из микропрограммного автомата поступает в АЛУ. Операционные блоки современных АЛУ строятся как комбинационные схемы, то есть они не обладают внутренней памятью и до момента сохранения результата операнды должны присутствовать на входе блока.

Регистры операндов

Регистры R_X и R_Y обеспечивают сохранение операндов на входе операционного блока вплоть до получения результата операции и его записи (в нашем случае в аккумулятор).

Регистр признаков

предназначен для фиксации и хранения признаков (флагов), характеризующих результат последней выполненной арифметической или логической операции. Такие признаки могут информировать о равенстве результата нулю, о знаке результата, о возникновении переноса из старшего разряда, переполнении разрядной сетки и т. д. Содержимое R_{Прз} обычно используется устройством управления для реализации условных переходов по результатам операций АЛУ. Под каждый из возможных признаков отводится один разряд R_{Прз}.

Аккумулятор (Акк)

это регистр, на который возлагаются самые разнообразные функции. Так, в него предварительно загружается один из операндов, участвующих в арифметической или логической операции. В Акк может храниться результат предыдущей команды и в него же заносится результат очередной операции. Через Акк зачастую производятся операции ввода и вывода.

Основная память

Вне зависимости от типа используемых микросхем основная память (ОП) представляет собой массив запоминающих элементов (ЗЭ), организованных в виде ячеек, способных хранить некую единицу информации, обычно один байт. Каждая ячейка имеет уникальный адрес. Ячейки ОП организованы в виде матрицы, а выбор ячейки осуществляется путем подачи разрешающих сигналов на соответствующие строку и столбец этой матрицы. Это обеспечивается дешифратором адреса памяти, преобразующим поступивший из РАП адрес ячейки в разрешающие сигналы, подаваемые в горизонтальную и вертикальную линии, на пересечении которых расположена адресуемая ячейка. При современной емкости ОП для реализации данных сигналов приходится использовать несколько микросхем запоминающих устройств (ЗУ). В этих условиях процесс обращения к ячейке состоит из выбора нужной микросхемы (на основании старших разрядов адреса) и выбора ячейки внутри микросхемы (определяется младшими разрядами адреса). Первая часть процедуры производится внешними схемами, а вторая — внутри микросхем ЗУ.

Модуль ввода/вывода

Структура приведенного *модуля ввода/вывода* (МВВ) обеспечивает только пояснение логики работы ВМ. В реальных ВМ реализация этого устройства машины может существенно отличаться от рассматриваемой. Задачей МВВ является обеспечение подключения к ВМ различных периферийных устройств (ПУ) и обмена информацией с ними. В рассматриваемом варианте МВВ состоит из дешифратора номера порта ввода/вывода, множества портов ввода и множества портов вывода.

Порты ввода и порты вывода

Портом называют схему, ответственную за передачу информации из периферийного устройства ввода в аккумулятор АЛУ (порт ввода) или из аккумулятора на периферийное устройство вывода (порт вывода). Схема обеспечивает электрическое и логическое сопряжение ВМ с подключенным к нему периферийным устройством.

Дешифратор номера порта ввода/вывода

В модуле ввода/вывода данной условной ВМ предполагается, что каждое ПУ подключается к своему порту. Каждый порт имеет уникальный номер, который указывается в адресной части команд ввода/вывода. *Дешифратор номера порта ввода/вывода* (ДВВ) обеспечивает преобразование номера порта в сигнал, разрешающий операцию ввода или вывода на соответствующем порте. Непосредственно ввод (вывод) происходит при поступлении из МП А сигнала Вв (Выв).