

## **ЛЕКЦИЯ 5-8**

**Логические основы компьютера, элементы и узлы  
Базовые представления об архитектуре ПК  
Процессор, структура и функционирование**

**Алгебра логики и логические элементы.** Для описания логики функционирования аппаратных и программных средств компьютера используется алгебра логики, или булева алгебра, британский математик Джордж Буль (1815-1846). Элементы алгебры логики, ее константы - ложь и истина (false и true), т.е. логический 0 и логическая 1. Алгебра логики оперирует логическими переменными, которые могут принимать только два значения - истина и ложь. Логической функцией  $F$  от набора логических переменных  $x_1, \dots, x_n$  называется функция, которая может принимать только два значения: логический 0 и логическая 1. Ранее считалось, что работы Д.Буля никому не нужны, и их автор подвергался насмешкам. Однако в 1938 году американский инженер Клод Шеннон положил булеву алгебру в основу теории электрических и электронных переключательных схем - сумматоров, создание которых и привело к появлению ЭВМ, способных автоматически производить арифметические вычисления. В результате алгебра логики явилась математической основой теории электрических и электронных переключательных схем и последующей разработки цифровых электронных вычислительных машин и микропроцессорных систем управления.

Прежде всего булева алгебра была использована для преобразования релейно-контактных схем. Если имеется электрическая цепь с параллельно соединенными контактами (рис 1 а), то ток в такой цепи будет протекать при замкнутом контакте А или В (или оба вместе). Если же контакты включены последовательно (рис.1 б), то ток в цепи потечет только тогда, когда замкнут контакт А и замкнут контакт В. Техническая реализация операции НЕ с помощью контактной схемы показана на рисунке 1 в).

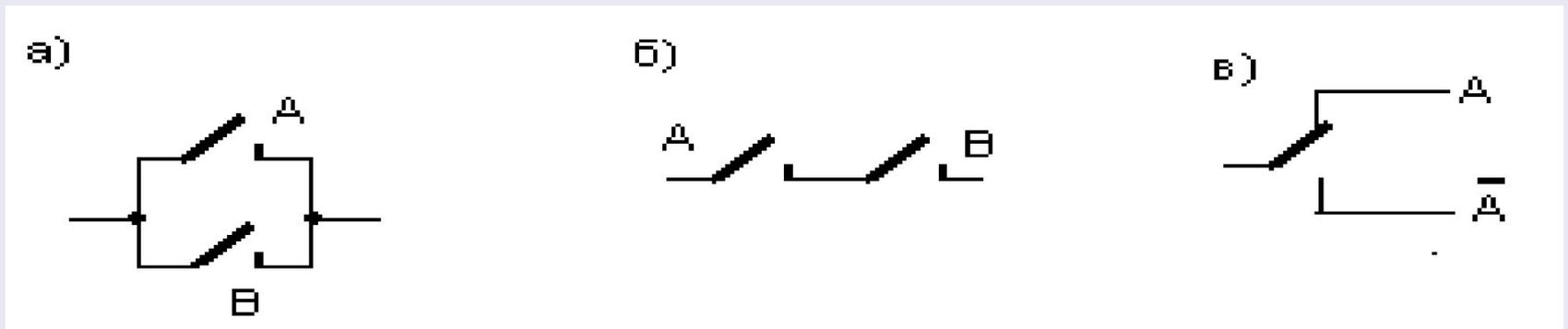


Рис. 1 - Реализация булевских функций контактными схемами: а - операция ИЛИ; б - операция И; в - операция НЕ

Обозначив замкнутые контакты - 1, а разомкнутые - 0, можно исследовать условия проводимости этих схем в зависимости от состояния контактов:

Сравнение этих таблиц с таблицами истинности для логических функций ИЛИ, И, НЕ показывают их полную идентичность.

Техническое устройство, реализующее логическую функцию, может рассматриваться просто как логический элемент, внутренняя структура которого не конкретизируется.

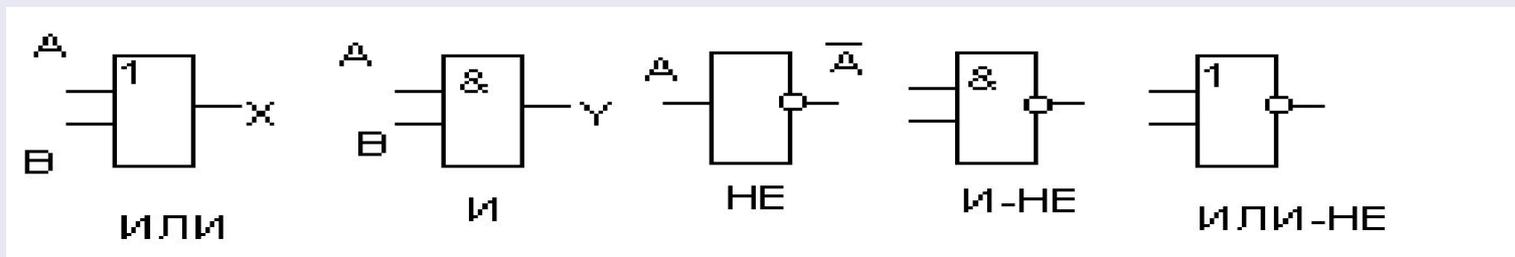


Рис. 2 - Схемные логические элементы вычислительных машин

Реальная аппаратура строится из логических элементов подобно тому, как сложная логическая функция получается путем комбинации более простых функций. Математическим аппаратом построения является булева алгебра. Она обеспечивает удобное средство представления различных устройств, не рисуя схемы соединений между элементами.

Логические схемы НЕ обычно реализуются на транзисторах, которые не только инвертируют входной сигнал, но и могут его усиливать. Обычно эти функции совмещаются и логические элементы НЕ, кроме выполнения логической операции, используются для восстановления уровня входного сигнала до заданного значения. Обычно уровень логической 1 составляет 3,3 В (в ранних версиях до Pentium, было +5 В), а уровень логического 0 - 0 В. В интегральных схемах логические элементы помещены в одном корпусе или даже на одном полупроводниковом кристалле со схемами усиления, которые одновременно производят инвертирование. В результате интегральные микросхемы реализуют не элементарные, а более сложные логические функции И-НЕ, либо ИЛИ-НЕ в зависимости от типа использованной логики (называемые обычно *вентильями*). Логическая функция И-НЕ, которая представляет собой отрицание логического умножения, называется операцией Шеффера. Логическая функция ИЛИ-НЕ, т.е. отрицание логического сложения, носит название «Стрелка Пирса» и обозначается как  $AB = A \downarrow B$

*Триггер* - электронная схема, широко применяемая в регистрах компьютера для запоминания одного разряда двоичного кода. Триггер имеет два устойчивых состояния, одно из которых соответствует двоичной единице, а другое - двоичному нулю. Термин триггер происходит от английского слова trigger - защелка, спусковой крючок. Это название указывает на способность мгновенно переходить «перебрасываться» из одного состояния в другое и обратно. Самый распространенный тип триггера RS-триггер (R и S соответственно от set - установка и reset - сброс). Условное обозначение приводится на рисунке 4.

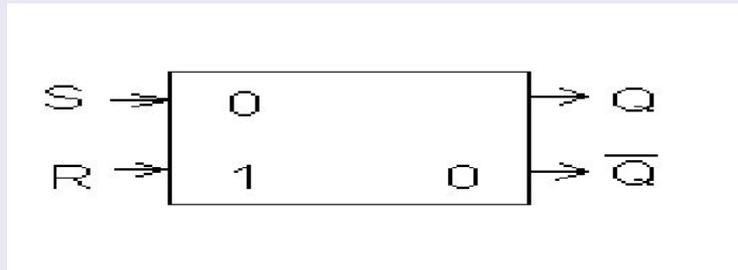


Рис. 4-RS-триггер

Интегральные схемы, реализующие операции И-НЕ и ИЛИ-НЕ, называют базовыми элементами микроэлектроники.

## Основные параметры логических элементов.

Логические элементы характеризуются многими параметрами. Среди них основными являются:

- *коэффициент объединения по входу* определяет число входов элемента, предназначенных для подачи логических переменных. Элемент с большим числом входов имеет более широкие функциональные возможности.
- *Нагрузочная способность (коэффициент разветвления по выходу)* определяет число входов аналогичных элементов, которое может быть подключено к выводу данного элемента. Чем выше коэффициент, тем меньше число элементов может потребоваться при построении цифрового устройства.
- *Быстродействие* является одним из важнейших его параметров. Оценивается *задержкой распространения сигнала* от входа к выходу элемента.
- *Помехоустойчивость* определяется максимальным значением помехи, не вызывающей нарушения работы элемента.

**Узлы компьютера.** Построение компьютера ведется по следующей цепочке: элементы -> узлы -> устройства.

**Узел** - это совокупность функционально связанных между собой элементов. Узлами ЭВМ являются стандартизированные наборы логических элементов, из которых, как из «кирпичиков», набираются схемы, входящие в состав микропроцессоров, блоков памяти, контроллеров внешних устройств и пр.

По способу функционирования узлы разделяются на:

- *комбинационные*, или узлы, выходные сигналы которых определяются только сигналом на входе, действующим в настоящий момент времени (например, дешифратор). Выходной сигнал дешифратора зависит только от двоичного кода, поданного на вход в настоящий момент времени. Комбинационные узлы называют также автоматами без памяти;

- *последовательностные* (автоматы с памятью) – это узлы, выходной сигнал которых зависит не только от комбинации входных сигналов, действующих в настоящий момент времени, но и от предыдущего состояния узла (счетчик).

Среди логических узлов ЭВМ широкое распространение получили комбинационные устройства (схемы). Такой автомат в общем случае представляется в виде многополюсника, имеющего  $R$  входов  $m_1, m_2, \dots, m_r$  и  $l$  выходов  $k_1, k_2, \dots, k_l$ . Поступающая на вход автомата информация задается набором сигналов  $M (m_1, m_2, \dots, m_r)$  образующим входное слово. При этом в любой дискретный момент времени  $t_i$  совокупность выходных сигналов – выходное слово  $K (k_1, k_2, \dots, k_l)$  – полностью определяется входным словом  $M$ , поступившим в тот же момент времени. При изменении набора входных сигналов  $M$  меняется набор выходных сигналов  $K$ .

Таким образом, выходные сигналы *комбинационного* аппарата полностью определяются входными сигналами и не зависят от внутреннего состояния автомата. Эти автоматы не имеют памяти и их называют автоматом без памяти.

В *последовательностных* устройствах (или автоматах с памятью) входной сигнал определяется не только набором символов, действующих на входах в данный момент времени, но и внутренним состоянием устройства, а последнее зависит от того, какие наборы действовали на входах во все предшествующие моменты времени в процессе работы устройства. Поэтому о них говорят, что устройства имеют память.

Узлы компьютера по своему функциональному назначению делятся на группы: регистры; дешифраторы; шифраторы; схемы сравнения кодов; электронные счетчики; сумматоры.



**О понятии «архитектура ЭВМ».**  
«Архитектура ЭВМ - это наиболее общие принципы построения ЭВМ, реализующие программное управление работой и взаимодействием основных ее функциональных узлов, на уровне, достаточном для понимания принципов работы и системы команд ЭВМ».

## Классическая архитектура ЭВМ и принципы фон Неймана.

Основы учения об архитектуре вычислительных машин заложил Джон фон Нейман (1903-1957). В процессе работы со своими коллегами Г. Голдстайном и А. Берксом фон Нейман высказал идею принципиально новой ЭВМ. В 1946 г. ученые изложили свои принципы построения вычислительных машин в статье «Предварительное рассмотрение логической конструкции электронно-вычислительного устройства». В статье обосновывается использование двоичной системы для представления чисел (ранее все вычислительные машины хранили обрабатываемые числа в десятичном виде). Авторы продемонстрировали преимущества двоичной системы для технической реализации, удобство и простоту выполнения в ней арифметических и логических операций. В дальнейшем ЭВМ стали обрабатывать и нечисловые виды информации - текстовую, графическую, звуковую и другие, но двоичное кодирование данных по-прежнему составляет информационную основу любого современного компьютера.

*Первый принцип* – информация кодируется в двоичной форме и разделяется на единицы – слова.

**Принцип адресности.** Структурно основная память состоит из перенумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Отсюда следует возможность давать имена областям памяти так, чтобы к запомненным в них значениям можно было впоследствии обращаться или менять их в процессе выполнения программ с использованием присвоенных имен – адресами слов.

Революционной идеей, является *принцип однородности памяти* «хранимой программы» и *принцип программного управления*. Первоначально программа задавалась путем установки переключателей на специальной коммутационной панели. Это было весьма трудоемким занятием: например, для изменения программы машины ENIAC требовалось несколько дней (в то время как собственно расчет не мог продолжаться более нескольких минут - выходили из строя лампы). *Нейман первым догадался, что программа состоит из набора команд, и может также храниться в виде набора нулей и единиц, причем в той же самой памяти, что и обрабатываемые ею числа.* Отсутствие принципиальной разницы между программой и данными дало возможность ЭВМ самой формировать для себя программу в соответствии с результатами вычислений. На этом принципе основаны методы трансляции – перевода текста программы с языка программирования высокого уровня на язык конкретной машины. Выборка программы из памяти осуществляется с помощью *счетчика команд (СК)*. Этот регистр процессора последовательно увеличивает хранимый в нем адрес очередной команды на длину команды.

***Основной алгоритм работы процессора:*** Начало цикла

- чтение команды из памяти по адресу, записанному в СК
- увеличение СК на длину прочитанной команды
- выполнение прочитанной команды . Конец цикла

Так как команды программы расположены в памяти друг за другом, то тем самым организуется выборка цепочки команд из последовательно расположенных ячеек памяти. Если после выполнения команды следует перейти не к следующей, а к какой-то другой, используются команды *условного или безусловного переходов (ветвления)*, которые заносят в счетчик команд номер ячейки памяти, содержащей следующую команду. Выборка прекращается после достижения и выполнения команды «стоп». Таким образом, процессор исполняет программу автоматически, без вмешательства человека.

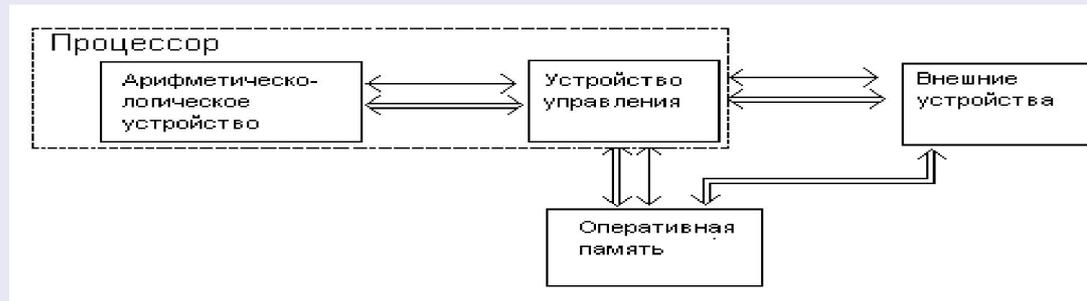
Таким образом, в основу построения цифровых вычислительных машин положены принципы: - использование 2-ичной системы счисления в машинной арифметике; адресуемость памяти; хранение данных и программ в общей памяти ЭВМ. Джон фон Нейман описал состав устройств *однопроцессорной ЭВМ и состав системы команд процессора*. Поэтому классическую архитектуру часто называют «архитектурой фон Неймана»



Так как команды программы расположены в памяти друг за другом, то тем самым организуется выборка цепочки команд из последовательно расположенных ячеек памяти. Если после выполнения команды следует перейти не к следующей, а к какой-то другой, используются команды *условного или безусловного переходов (ветвления)*, которые заносят в счетчик команд номер ячейки памяти, содержащей следующую команду. Выборка прекращается после достижения и выполнения команды «стоп». Таким образом, процессор исполняет программу автоматически, без вмешательства человека.

Таким образом, в основу построения цифровых вычислительных машин положены принципы: - использование 2-ичной системы счисления в машинной арифметике; адресуемость памяти; хранение данных и программ в общей памяти ЭВМ. Джон фон Нейман описал состав устройств однопроцессорной ЭВМ и состав системы команд процессора. Поэтому классическую архитектуру часто называют «архитектурой фон Неймана»

Фон Нейман не только выдвинул основополагающие принципы логического устройства ЭВМ, но и предложил ее структуру, которая воспроизводилась в течение первых двух поколений ЭВМ. Основными блоками по Нейману являются устройство управления (УУ) и арифметико-логическое устройство (АЛУ), память (ОЗУ), внешняя память (ВЗУ), устройства ввода и вывода. Следует отметить, что внешняя память отличается от устройств ввода и вывода тем, что данные в нее заносятся в виде, удобном компьютеру, но недоступном для непосредственного восприятия человеком. Так, накопитель на магнитных дисках относится к внешней памяти, а клавиатура - устройство ввода, дисплей и принтер - устройства вывода.



с. 4.1 - Структура фон Неймана

Устройство управления и арифметико-логическое устройство в современных компьютерах объединены в один блок - процессор, являющийся преобразователем информации, поступающей из памяти и внешних устройств (сюда относятся выборка команд из памяти, кодирование и декодирование, выполнение различных, в том числе и арифметических, операций, согласование работы узлов компьютера).



Память (ЗУ) хранит информацию (данные) и программы. Запоминающее устройство у современных компьютеров «многоярусно» и включает:

оперативное запоминающее устройство (ОЗУ), хранящее ту информацию, с которой компьютер работает непосредственно в данное время (исполняемая программа, часть необходимых для нее данных, некоторые управляющие программы), внешние запоминающие устройства (ВЗУ) гораздо большей емкости, чем ОЗУ, но с существенно более медленным доступом (и значительно меньшей стоимостью в расчете на 1 байт хранимой информации).

На ОЗУ и ВЗУ классификация устройств памяти не заканчивается - определенные функции выполняют и СОЗУ (сверхоперативное запоминающее устройство - кэш-память CacheMemory), и ПЗУ (постоянное запоминающее устройство), и другие подвиды компьютерной памяти.

В построенной по описанной схеме ЭВМ происходит последовательное считывание команд из памяти и их выполнение. Номер (адрес) очередной ячейки памяти, из которой будет извлечена следующая команда программы, указывается специальным устройством - счетчиком команд в УУ. Его наличие также является одним из характерных признаков рассматриваемой архитектуры. Разработанные фон Нейманом основы архитектуры вычислительных устройств оказались настолько фундаментальными, что получили в литературе название «фон-неймановской архитектуры». Исключение составляют лишь отдельные разновидности систем для параллельных вычислений, в которых отсутствует счетчик команд, не реализована классическая концепция переменной и имеются другие существенные принципиальные отличия от классической модели (примерами могут служить потоковая и редуцирующая вычислительные машины).

**Типы архитектур.** Основные классические типы архитектур можно определить как: «звезда», иерархическая, магистральная.

**Архитектура «звезда»** - ЦУ соединено непосредственно с ВУ и управляет их работой (ранние модели машин). Классическая архитектура – одно арифметико-логическое устройство (АЛУ), через которое проходит поток данных, и одно устройство управления (УУ), через которое проходит поток команд – программа. Это однопроцессорный компьютер.

Вычислительная машина включает пять базовых компонент и состоит из следующих типов устройств: центральный процессор (ЦП), включающий АЛУ и УУ; запоминающие устройства – память, в том числе оперативная (ОП) и внешние ЗУ; устройства ввода и устройства вывода информации – внешние (периферийные) устройства (ВУ).

**Иерархическая архитектура** – ЦУ соединено с периферийными процессорами (вспомогательными процессорами, каналами и пр.), управляющими контроллерами, к которым подключены группы ВУ.

**Магистральная структура** – процессор (процессоры) и блоки памяти (ОП) взаимодействует между собой и с ВУ (контроллерами ВУ) через внутренний канал, общий для всех устройств (машины ПЭВМ IBM, PC-совместимые). К этому типу архитектуры относится также архитектура ПК: -функциональные блоки связаны между собой общей шиной (магистралью). Физически *магистраль* представляет собой многопроводную линию с гнездами для подключения электронных схем. Совокупность проводов магистрали разделяется на отдельные группы: шину адреса, шину данных и шину управления.



Периферийные устройства подключаются к аппаратуре компьютера через **специальные контроллеры** – устройства управления периферийными устройствами.

Контроллер – устройство, которое связывает периферийное оборудование или каналы связи с ЦП, освобождая процессор от непосредственного управления функционированием данного оборудования.

Все рассмотренные архитектуры базируются на следующих схемных элементах и базовых узлах: память обычно использует возможности и свойства триггера или его аналогов; счетчик (регистр) адреса команд, сумматор – полный сумматор или полусумматор; дешифратор.

## Процессор, структура и функционирование. **Абстрактное центральное устройство.**

Перечислим основные понятия и рассмотрим структуру и функции абстрактного центрального устройства ЭВМ (1-2-е поколения ЭВМ). *Команда* – описание операции, которую необходимо выполнить. Каждая команда начинается с *кода операции* (КОП), содержит необходимые *адреса*, характеризуется форматом, который определяет структуру команды, ее организацию, код, длину, метод расположения адресов. Длина различных команд может быть как одинаковой, так и разной. Число разрядов в КОП определяется количеством различных операций производимой той или иной машиной и связывается формулой:  $M=2^n$ , где  $M$  – количество операций,  $n$  – число разрядов в КОП. Команды подразделяются на арифметические, логические, ввода/вывода, передачи данных. Каждая команда выполняется в компьютере за один либо несколько *тактов*. Последовательность взаимосвязанных команд именуется *макрокомандой*. Использование макрокоманд упрощает программирование. Цикл процессора – период времени, за который осуществляется выполнение команды исходной программы в машинном виде; состоит из нескольких *тактов*. Такт работы процессора – промежуток времени между соседними импульсами *генератора тактовых импульсов*, частота которых есть *тактовая частота процессора*. Эта частота является одной из основных характеристик компьютера и во многом определяет скорость его работы, т.к. каждая операция выполняется за определенное количество тактов. Выполнение *короткой команды* (арифметика с фиксированной точкой, логические операции) обычно занимает пять тактов: выборка команды; расшифровка кода операции (декодирование); генерация адреса и выборка данных из памяти; выполнение операции; запись результата в память.

*Регистры* – устройства, предназначенные для временного хранения данных ограниченного размера. Важной характеристикой регистра является высокая скорость приема и выдачи данных. Регистр состоит из разрядов, в которые можно быстро записывать, запоминать и считывать слово, команду, двоичное число и т.д. Обычно регистр имеет ту же разрядность, что и машинное слово. Регистр, накапливающий данные, именуют *аккумулятором*. Регистр, обладающий способностью перемещать содержимое своих разрядов, называют *сдвиговым*. В этих регистрах за один такт хранимое слово поразрядно сдвигается на одну позицию. Сдвиговые регистры используются при обработке данных, кодировании и декодировании. Некоторые регистры служат *счетчиками*. Счетчик является устройством, которое на своих выходах выдает (в двоичной форме) сумму числа импульсов, подаваемых на его единственный вход. Максимальное число импульсов, которое счетчик может подсчитать, называется его *емкостью*.

*Регистры общего назначения (РОН)* – общее название для регистров, которые временно содержат данные, передаваемые или принимаемые из памяти. *Регистр команды (РК)* – служит для размещения текущей команды, которая находится в нем в течение текущего цикла процессора. Регистр (РАК), счетчик (СЧАК) адреса команды – регистр, содержащий адрес текущей команды. Регистр адреса (числа) - РА(Ч) – содержит адрес одного из операндов выполняемой команды (регистров может быть несколько).

**Командный цикл процессора.** Цикл выполнения команды может выглядеть следующим образом.

1. В соответствии с содержимым СчАК (адрес очередной команды) УУ извлекает из ОП очередную команду и помещает ее в РК (чтение команды).

Команда содержит: КОП – код операции (сложение, вычитание, сравнение, обращение к ВУ и пр.). Номера регистров (в некоторых машинах – адреса слов, ячеек ОП) в которых размещена соответствующая информация. Адреса операндов А1, А2 и т.д. участвующих в выполнении команды (чисел, строк, других команд программы).

2. Осуществляется расшифровка (декодирование) команды.

3. Адреса А1, А2 и пр. помещаются в регистры адреса (РА).

4. По значениям РА осуществляется чтение содержимого и помещение его в РЧ.

5. Выполнение операции и помещение результата в РР.

Запись результата по одному из адресов.

Увеличение содержимого СчАК на единицу (переход к следующей команде).

Очевидно, что за счет увеличения числа регистров возможно *распараллеливание, перекрытие* операций. Все это является предпосылкой построения *конвейерных* структур.

**Процессор.** Под *архитектурой* процессора понимается его программная модель, то есть программно-видимые свойства. *Программная модель процессора* – это функциональная модель, используемая программистом при разработке программ в кодах ЭВМ или на языке ассемблера.

Под *микроархитектурой* понимается аппаратная реализация этой программной модели. В состав процессора входят арифметико-логическое устройство (АЛУ), устройство управления (УУ) и группа регистров.

**Арифметико-логическое устройство (АЛУ).** Arithmetic and Logical Unit (ALU) - компонента процессора, которая осуществляет арифметические и логические операции. АЛУ обеспечивает выполнение основных операций по обработке информации. Основу АЛУ составляет сумматор с последовательно-параллельным переносом.

**Сумматор** - электронная схема, используемая для сложения двоичных чисел. АЛУ классифицируются:

1. по способу действий над операндами

- последовательного действия;
- параллельного действия.

2. по виду обрабатываемых чисел.

3. по организации действий над операндами:

- блочные (одни предназначены для действий над двоично-десятичными числами, другие для чисел с фиксированной запятой, для чисел с плавающей запятой);
- многофункциональные (одни и те же блоки обрабатывают числа в любой форме представления)

4. по структуре:

- с непосредственными связями;
- многосвязные. Любую задачу компьютер разбивает на отдельные логические операции, производимые над двоичными числами, причем в одну секунду осуществляются сотни тысяч или миллионы таких операций. Сложение, вычитание, умножение и деление - элементарные операции, выполняемые АЛУ ЭВМ. Полный набор таких операций называют системой команд, а схемы их реализации составляют основу АЛУ.

Помимо арифметического устройства, АЛУ включает и логическое устройство, предназначенное для операций, при осуществлении которых отсутствует перенос из разряда в разряд. Иногда эти операции называют «логическое И» «и логическое ИЛИ». Все операции в АЛУ производятся в регистрах – специально отведенных ячейках АЛУ. Время выполнения простейших операций определяется минимальным временем сложения двух операндов, находящихся в регистрах. В случае, если одно или оба слагаемых находятся не в регистрах, а в оперативном запоминающем устройстве (ОЗУ), учитывается также время пересылки слагаемых в регистры и время записи полученной суммы в ОЗУ. В большинстве современных микропроцессоров это время составляет от нескольких сотен наносекунд до нескольких микросекунд.

*Устройство управления (УУ)* – формирует и подает во все блоки процессора в нужные моменты времени определенные сигналы управления, обусловленные спецификой выполняемой операции и результатами предыдущих операций; формирует адреса ячеек памяти, используемых выполняемой операцией, и передает эти адреса в соответствующие блоки ЭВМ. Устройство управления содержит регистр команд, дешифратор команд и управляющее устройство, в состав которого входит управляющая память. Управляющая память содержит микропрограммы всех команд для данного МП. На управляющее устройство поступают последовательности тактового генератора, сигнал готовности от ОЗУ, постоянного запоминающего устройства (ПЗУ) или устройств ввода-вывода (УВВ) к приему или передаче данных, сигнал запроса на прерывание от внешних устройств.

**Системы команд и соответствующие классы процессоров.** Основные команды ЭВМ классифицируются: по функциям (по выполняемым операциям), направлению приема-передачи информации и адресности.

Классы команд

1. Команды обработки данных:

Короткие: логические; арифметические; длинные: сложение/вычитание с фиксированной точкой; умножение/деление с фиксированной точкой.

Операции управления: безусловный переход; условный переход.

Операции обращения к внешним устройствам.

Команды по типу выборки и пересылки данных: регистр-регистр (операнды размещаются в регистрах АЛУ); память-регистр или регистр-память; память-память (операнды размещены в ОП).

Известны одно-, двух- и трехадресные машины (системы команд). В трехадресной машине сложение двух чисел требует одной команды (извлечь число по А1, число по А2, сложить и записать результат по А3). В двухадресной необходимы две команды (извлечь число по А1 и поместить в РЧ (или сумматор), извлечь число по А1, сложить с содержимым РЧ и результат записать по А2). В одноадресной машине потребуется 3 команды. Поэтому основная тенденция в развитии ЦУ ЭВМ состоит в увеличении разрядности.

Процесс выполнения команды состоит из двух этапов: выборка и выполнение. Цикл процессора – цикл выборки и выполнения команды.

Выборка начинается со считывания из счетчика команд номера ячейки ОЗУ, содержащей код команды.

После считывания содержимое счетчика команд сразу увеличивается на 1. Номер ячейки ОЗУ передается через регистр адреса памяти и адресную шину в дешифратор ОЗУ.

Дешифратор ОЗУ выбирает ячейку ОЗУ, содержащую код команды. Код команды считывается из ОЗУ и через шину данных передается в регистр данных памяти. Из регистра данных памяти код команды передается в регистр команд, где он хранится до конца выполнения команды, и передается в АЛУ. АЛУ анализирует код команды и, если не нужно дополнительного обращения к памяти, переходит к выполнению.

Если же нужно дополнительное обращение к памяти, то МП переходит ко второму машинному циклу, который также начинается с выборки. МП запрашивает в ОЗУ дополнительные данные и выполняет команды.

Команды могут выполняться за 1 или несколько машинных циклов. В каждом машинном цикле происходит только одно обращение к памяти. Выполнение команды происходит под управлением сигналов, вырабатываемых устройством управления. При выполнении команды АЛУ взаимодействует с РОН. РОН используются для кратковременного хранения операндов и результатов.

Затраты времени на выполнение одной команды можно определить, умножая число тактов синхронизации, необходимых для выполнения команды, на период синхронизации. Это время можно выразить в виде суммы базового времени выполнения (которое зависит от команды и режима адресации) и времени вычисления эффективного адреса, если привлекается операнд из памяти. Базовое время выполнения предполагает, что выполняемая команда уже выбрана и находится в очереди команд. В противном случае требуется учесть дополнительные такты синхронизации, необходимые для выборки команды.

*Классы процессоров.* В зависимости от набора и порядка выполнения команд процессоры подразделяются на 4 класса, отражающих также последовательность развития ЭВМ.

Ранее других появились процессоры CISC. Затем, с целью повышения быстродействия процессоров были разработаны процессоры RISC, которые характеризуются сокращенным набором быстро выполняемых команд. Позже появилась концепция процессоров MISC, использующая минимальный набор длинных команд. Вслед за ними возникли процессоры VLIW, работающие со сверхдлинными командами.

CISC (Complete Instruction Set Computer) – традиционная архитектура, в которой ЦП использует микропрограммы для выполнения полного набора команд микропроцессора. Они могут иметь разную длину, методы адресации и требуют сложных электронных цепей для декодирования и исполнения. В 1974 г. John Cocke (IBM Research) предложил новый подход, который мог существенно уменьшить количество машинных команд в ЦП.

RISC (Reduced (Restricted) Instruction Set Computer) – процессор, функционирующий с сокращенным набором команд, и содержащий только наиболее простые команды. В результате аппаратная реализация такой архитектуры позволяет с небольшими затратами декодировать и выполнять эти инструкции за минимальное число тактов синхронизации. Определенные преимущества дает и унификация регистров. Первый процессор RISC был создан корпорацией в 1979 г. и имел шифр IBM 801. Главными идеями этих машин было отделение медленной памяти от высокоскоростных регистров и использование регистровых окон. Эти три процессора имели много общего. Все они придерживались архитектуры, отделяющей команды обработки от команд работы с памятью, и делали упор на эффективную конвейерную обработку.



Процессор MISC – процессор, работающий с минимальным набором длинных команд. Увеличение разрядности процессоров привело к идее укладки нескольких команд в одно слово (связку) размером 128 бит. Оперируя одним словом, процессор получил возможность обрабатывать сразу несколько команд. Это позволило использовать возросшую производительность компьютера и его возможность обрабатывать одновременно несколько потоков данных.

Процессор VLIW - процессор, работающий с системой команд сверх большой разрядности. Идея технологии VLIW заключается в том, что создается специальный компилятор планирования, который перед выполнением программы проводит ее анализ, и по множеству ветвей последовательности операций определяет группу команд, которые могут выполняться параллельно. Каждая такая группа образует одну сверхдлинную команду. Это позволяет решить две задачи. Во-первых, в течение одного такта выполнить группу коротких команд, во-вторых, упростить структуру процессора. Технология появилась в результате работ, проведенных корпорациями HP, Intel.

# Литература

1. Вычислительные системы, сети и телекоммуникации. Пятибратов и др. - ФИС, 2000.
2. Сергеев Н.П., Вашкевич Н.П. Основы вычислительной техники. –М.:ВШ, 1988.
3. Гук М. Процессоры фирмы Intel от 8086 до PENTIUM II. -Санкт-Петербург: Питер-Пресс, 1998.
4. Гук М. Аппаратные средства IBM PC. –Санкт –Петербург: Питер, 2000.
5. Пухальский Г.И., Новосельцева Т.Я. Цифровые устройства: Учебное пособие для вузов. \_СПб.: Политехника, 1996. -866с.
6. Самофалов К.Г., Корнейчук И.В., Тарасенко В.П. Цифровые электронные вычислительные машины. –Киев: ВИЦА ШКОЛА, 1983.
7. Гук М., Юров В. PENTIUM 4 Athlon и Duron. –Санкт –Петербург: Питер, 2001.
8. Томпсон Р.Б., Томпсон Б.Ф. Железо ПК: Энциклопедия. –СПб.: Питер, 2003.
9. В.Л.Бройдо, О.П.Ильина. Архитектура ЭВМ и систем. Питер, 2006, 703с.
10. Таненбаум Э. Архитектура компьютера. СПб, Питер, 2005, 685с.
11. Мюллер М. Аппаратные средства ПК. М. Радио, 2004.
12. Утепбергенов, И.Т. Архитектура компьютерных систем: Учебное пособие. / Алматы: Экономика, 2010.- 265 с.