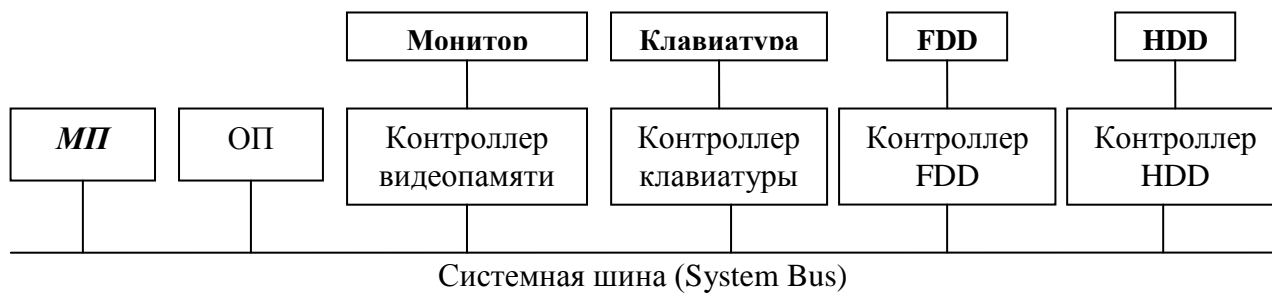


## 2 Раздел Общие принципы построения ОС

### 2.1 Машинно-зависимые свойства ОС

#### 2.1.1 Архитектурные особенности модели микропроцессорной системы



#### Микропроцессор

Первые процессоры появились на самой заре зарождения компьютерных технологий. А бурное развитие микрокомпьютерной техники во многом являлось следствием появления первых *микропроцессоров*. Если раньше все необходимые элементы CPU были расположены на различных электронных схемах, то в микропроцессорах они впервые были объединены на одном-единственном кристалле. В дальнейшем под термином «процессор» мы будем иметь в виду именно микропроцессоры, поскольку эти слова давно превратились в синонимы.



Микропроцессор i4004 — прадедушка сегодняшних CPU

Одним из первых микропроцессоров был четырехразрядный процессор фирмы Intel i4004. Он имел смехотворные по нынешним временам характеристики, но для своего времени — начала 1970-х гг., его появление представляло собой настоящий технологический прорыв. Как можно догадаться из его обозначения, он был четырехразрядным и имел тактовую частоту около 0,1 МГц. И именно его прямой потомок, процессор i8088, был выбран фирмой IBM в качестве «мозга» первого персонального компьютера фирмы IBM PC.



Процессор i8088 использовавшийся в первом персональном компьютере фирмы IBM

Шли годы, характеристики CPU становились все более серьезными и внушительными, и, как следствие, становились все более солидными характеристики персональных компьютеров. Значительной вехой в развитии микропроцессоров стал i80386. Это был первый полностью 32-разрядный CPU, который мог адресовать к 4 ГБ оперативной памяти, в то время как большинство его предшественников могло работать максимум с

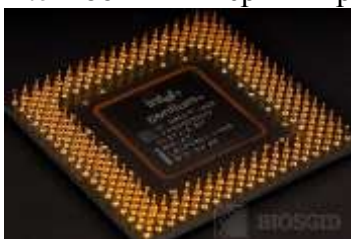
640 КБ ОЗУ. Подобная разрядность микропроцессоров настольных компьютеров продержалась довольно долго, почти два десятилетия. В середине 80-х объем ОЗУ в 4 ГБ казался фантастически огромным, но сейчас его можно считать небольшим для серьезного компьютера.



i80386 — первый полностью 32-разрядный CPU

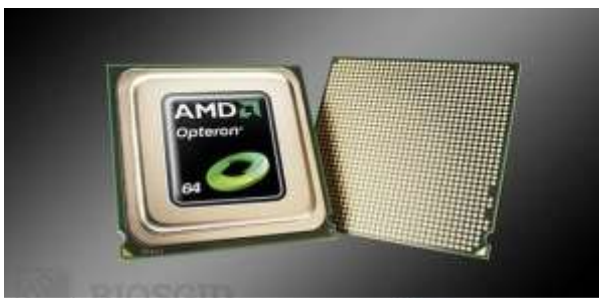
Следующий микропроцессор компании Intel, 486DX, замечателен тем, что в нем впервые появился внутренний кэш – внутренняя оперативная память микропроцессора. Кроме того, в нем было применено много других усовершенствований, которые во многом определили дальнейшую эволюцию микропроцессоров. То же самое можно сказать и про следующий процессор компании Intel, Pentium.

Intel 486DX — первый процессор с внутренним кэшем



Процессор компании Intel — Pentium

Вместе с CPU Pentium 4 в ряду технологий, использующихся в микропроцессорах, появилась технология Hyper Threading. А процессоры Opteron от фирмы AMD и Pentium D от Intel открыли современную эпоху эволюции CPU, эпоху процессоров, имеющих несколько ядер. Сейчас на рынке представлено много CPU от различных производителей, но главными производителями до сих пор остаются две компании – Intel и AMD, причем на долю первой приходится более 80% рынка.



CPU Opteron от фирмы AMD и Pentium D от Intel

## 2.2 Устройство CPU

Любой CPU имеет вычислительное ядро (иногда их бывает несколько), а также кэш, то есть собственную оперативную память. Кэш обычно имеет два уровня – первый и второй (внутренний и внешний). Внутренний имеет меньший объем, но обладает большим быстродействием по сравнению с внешним. Емкость кэша второго уровня современных CPU составляет несколько мегабайт – больше, чем оперативная память первых персональных компьютеров!

В ядре CPU находится несколько функциональных блоков – блок управления, блок выборки инструкций, блок вычислений с плавающей точкой, блок целочисленных вычислений, и т.д. Также в ядре располагаются главные регистры processor-a, в которых находятся обрабатываемые в определенный момент данные. В классической схеме микропроцессора архитектуры x86 этих регистров всего 16.

**На сегодняшний день наибольшее распространение получили две основные разновидности процессоров – CISC (Complex Instruction Set Computing) и RISC (Reduced Instruction Set Computing).** В CISC-процессорах мало внутренних регистров, но они поддерживают большой набор команд. В RISC-процессорах регистров много, зато набор команд ограничен. Традиционно микропроцессоры для персональных компьютеров архитектуры Intel x86 принадлежали к классу CISC-процессоров, однако в настоящее время большинство микропроцессоров представляют собой гибрид этих двух архитектур.

Если рассмотреть CPU на аппаратном уровне, то он является, по сути, огромной микросхемой, расположенной на цельном кристалле кремния, в которой содержатся миллионы, а то и миллиарды транзисторов. Чем меньше размеры транзисторов, тем больше их содержится на единицу площади CPU, и тем больше его вычислительная мощность. Кроме того, от размеров транзисторов зависит энерговыделение и энергопотребление процессора — чем меньше их размер, тем эти характеристики процессора меньше. Этот фактор немаловажен, так как CPU является наиболее энергоемким устройством современного ПК. Поэтому проблема уменьшения нагрева процессора входит в число самых важных, стоящих перед разработчиками ПК и самих процессоров.

Отдельно стоит сказать о корпусе, в котором находится CPU. Обычно материалом корпуса процессора служит керамика или пластик. Первоначально процессоры намертво впаивались в системную плату, сейчас же большинство вставляются в специальные гнезда – сокет. Такой подход заметно упростил модернизацию системы пользователем – достаточно вставить в разъем другой CPU, поддерживаемый данной системной платой, и вы получите более мощный компьютер.



Сокет современного процессора

С другими устройствами процессор связан при помощи специальных каналов связи (шин) – шины памяти, шины данных и шины адреса. Разрядность последней очень важна, поскольку от этого параметра зависит объем доступной CPU, а значит, и программ, оперативной памяти.

### 2.3 Принцип работы МП

**МП** включает в себя:

- 1) АЛУ
- 2) УУ
- 3) Регистры внутренней памяти – это устройство для запоминания определенного кода (32 и 64 битные)
- 4) КЭШ (cache) – это стековая память для увеличения быстродействия (128kb, 256kb, 512kb и т.д.)

Для обработки данных управляющее устройство CPU получает из оперативной памяти или кэша процессора сами данные, а также команды, которые описывают процесс обработки данных. Данные помещаются во внутренние регистры микропроцессора, и над ними производятся операции при помощи арифметико-логического устройства в соответствии с поступившими командами.



Принцип работы процессора

Работу CPU синхронизируют так называемые тактовые сигналы. Наверняка каждому пользователю известно понятие тактовой частоты, которая отражает количество тактов работы процессора за секунду. Это значение во многом определяет характеристики процессора. Тем не менее, производительность компьютера далеко не всегда пропорциональна его тактовой частоте. И дело тут не только в наличии у современных CPU нескольких ядер, а и в том, что разные процессоры имеют разную архитектуру и, как следствие, могут выполнять разное количество операций за секунду. Современные CPU могут выполнять несколько операций за один такт, тогда как у первых микропроцессоров на одну операцию, наоборот, могло уходить несколько тактов.

CPU архитектуры x86 исторически поддерживают следующие режимы работы процессора:

1. Реальный
2. Защищенный
3. Виртуальный
4. Режим супервизора

Этапы цикла выполнения:

1. Процессор выставляет число, хранящееся в [регистре счётчика команд](#), на [шину адреса](#) и отдаёт [памяти](#) команду чтения.
2. Выставленное число является для памяти [адресом](#); память, получив адрес и команду чтения, выставляет содержимое, хранящееся по этому адресу, на [шину данных](#) и сообщает о готовности.
3. Процессор получает число с шины данных, интерпретирует его как команду ([машинную инструкцию](#)) из своей [системы команд](#) и исполняет её.
4. Если последняя команда не является [командой перехода](#), процессор увеличивает на единицу (в предположении, что длина каждой команды равна единице) число, хранящееся в счётчике команд; в результате там образуется адрес следующей команды.

Данный цикл выполняется неизменно, и именно он называется *процессом* (откуда и произошло название устройства).

Во время процесса процессор считывает последовательность команд, содержащихся в памяти, и исполняет их. Такая последовательность команд называется [программой](#) и представляет [алгоритм](#) работы процессора. Очередность считывания команд изменяется в случае, если процессор считывает команду перехода, — тогда адрес следующей команды может оказаться другим. Другим примером изменения процесса может служить случай получения [команды останова](#) или переключение в режим обработки [прерывания](#).

Команды центрального процессора являются самым нижним уровнем управления компьютером, поэтому выполнение каждой команды неизбежно и безусловно. Не производится никакой проверки на допустимость выполняемых действий, в частности, не проверяется возможная потеря ценных данных. Чтобы компьютер выполнял только допустимые действия, команды должны быть соответствующим образом организованы в виде необходимой программы.

Скорость перехода от одного этапа цикла к другому определяется [тактовым генератором](#). Тактовый генератор вырабатывает импульсы, служащие ритмом для центрального процессора. Частота тактовых импульсов называется [тактовой частотой](#).

Реальный режим работы был единственным режимом, в котором работали все CPU до i80386. В этом режиме процессор мог адресовать лишь 640 КБ ОЗУ. В результате появления защищенного режима процессор получил возможность работать с большими объемами оперативной памяти. Также существует разновидность защищенного режима — виртуальный режим, предназначенный для совместимости со старыми программами, написанными для процессоров 8086.

Режимы работы процессора также включают режим супервизора, который используется при работе в современных операционных системах. В этом режиме программный код имеет неограниченный доступ ко всем системным ресурсам.

#### **МП выполняет команды следующим образом:**

- 1) осуществляет выборку команды из **ОЗУ** по адресу, который записан в счетчике адреса команд. При первоначальной загрузке программного кода в **ОЗУ** в счетчик команд записывается начальный адрес первой команды, таким образом, **МП** «знает» по какому адресу выбирать текущую команду.
- 2) Выбранная команда декодируется; в регистры внутренней памяти переписываются из **ОЗУ** переменные — операнды.
- 3) Команда исполняется **МП**, результат записывается в регистр временных результатов. Счетчик команд корректируется на длину (в байтах) команды и таким образом получается адрес следующей команды, к которой **МП** вновь обращается за выборкой





**МП** содержит регистр **PSW** – слово состояния процессора, в данный регистр записываются биты кода состояния, которые задают приоритет **МП** и режим (пользователя или ядра), длину команды (в байтах), которая выполняется в **МП** и другую служебную информацию. Пользовательские программы могут читать информацию из **PSW** целиком, но писать только в отдельные поля. Режим пользователя ограничивает область памяти (адреса в которые можно записывать информацию, а так же обращения к ПУ).

Рассмотренный выше процесс выполнения команды характеризует простую модель обработки. Современные **МП** обладают возможностями выполнения нескольких команд одновременно, т.е. во время выполнения  $n$ -ой команды **МП** может декодировать команду  $n+1$  и выбирать (считывать) команду  $n+2$ . Подобная ситуация называется конвейером, но при конвейерной модели усложняется создание компиляторов и системных программ.

Более передовой моделью является супер скалярный центральный процессор.

### 2.3.1 Конвейерная архитектура

Конвейерная архитектура ([англ. \*pipelining\*](#)) была введена в центральный процессор с целью повышения быстродействия. Обычно для выполнения каждой команды требуется осуществить некоторое количество однотипных операций, например: выборка команды из [ОЗУ](#), дешифровка команды, адресация операнда в ОЗУ, выборка операнда из ОЗУ, выполнение команды, запись результата в ОЗУ. Каждую из этих операций сопоставляют одной ступени конвейера. Например, конвейер микропроцессора с архитектурой [MIPS-I](#) содержит четыре стадии:

- получение и декодирование инструкции,
- адресация и выборка операнда из ОЗУ,
- выполнение арифметических операций,
- сохранение результата операции.

После освобождения  $k$ -й ступени конвейера она сразу приступает к работе над следующей командой. Если предположить, что каждая ступень конвейера тратит единицу времени на свою работу, то выполнение команды на конвейере длиной в  $n$  ступеней займёт  $n$  единиц времени, однако в самом оптимистичном случае результат выполнения каждой следующей команды будет получаться через каждую единицу времени.

Действительно, при отсутствии конвейера выполнение команды займёт  $n$  единиц времени (так как для выполнения команды по-прежнему необходимо выполнять выборку, дешифровку и т. д.), и для исполнения  $m$  команд понадобится  $n \cdot m$  единиц времени; при использовании конвейера (в самом оптимистичном случае) для выполнения  $m$  команд понадобится всего лишь  $n + m$  единиц времени.

Факторы, снижающие эффективность конвейера:

1. Простой конвейера, когда некоторые ступени не используются (например, адресация и выборка операнда из ОЗУ не нужны, если команда работает с регистрами).
2. Ожидание: если следующая команда использует результат предыдущей, то последняя не может начать выполняться до выполнения первой (это преодолевается при использовании внеочередного выполнения команд — out-of-order execution).
3. Очистка конвейера при попадании в него команды перехода (эту проблему удаётся сгладить, используя предсказание переходов).

Некоторые современные процессоры имеют более 30 ступеней в конвейере, что повышает производительность процессора, но, однако, приводит к увеличению длительности простоя (например, в случае ошибки в предсказании условного перехода). Не существует единого мнения по поводу оптимальной длины конвейера: различные программы могут иметь существенно различные требования.