EASY

* Представление символьных и строковых данных. Принципы построения кодовых таблиц ASCII, КОИ-8, ISO8859-5, Windows-1251, UTF-8, UTF-16.
* ~~Две формы представления информации. Способы представления дискретной информации. Системы счисления, используемые в вычислительной технике: двоичная, 8-я, 10-я, 16-я, двоично-десятичная.~~
* Представление чисел с фиксированной точкой. ~~Прямой, обратный и дополнительный код~~. Формирование битовых признаков переноса, переполнения, ~~отрицательного результата, нуля~~
* Организация вычислений в БЭВМ. Сдвиги, арифметические и логические операции. Цикл выборки команды.
* ~~Базовые элементы вычислительной техники: ячейки, регистры, шины, вентили, тактовые генераторы, логические схемы, триггеры, регистры, счетчики, сумматоры.~~
* ~~Состав и структура БЭВМ. Адресные пространства БЭВМ. Система команд БЭВМ, форматы команд. Машинные циклы.~~
* ~~Организация массивов данных~~. ~~Косвенная адресация~~. Цикл выборки адреса БЭВМ
* Управление вычислительным процессом в БЭВМ. Цикл исполнения команды ISZ.
* ~~Организация ввода-вывода в вычислительных системах. Инициация обмена, передача информации и завершение обмена. Драйверы.~~
* Организация асинхронного обмена в БЭВМ. Пример программы~~.~~ ~~Временные издержки асинхронного обмена~~.
* ~~Организация обмена по прерыванию программы в БЭВМ. Пример программы. Цикл прерывания~~
* ~~Организация ввода-вывода в БЭВМ. Устройства ввода-вывода, команды~~.
* Операционная система Unix — ядро ОС и файловая система.
* Операционная система Unix — интерпретаторы, стандартные потоки ввода вывода, фильтры.
* Операционная система Unix — основные команды, права файлов и способы их задания.

MEDIUM

\* ~~Структура и принцип функционирования ЭВМ. Порядок функционирования простого процессора на~~

~~примере калькулятора.~~

\* Подпрограммы в БЭВМ. Передача параметров. Цикл исполнения команд перехода и возврата из

подпрограммы. Стек, передача параметров. Позиционно-независимый код. Загрузчик и библиотеки.

\* ~~Понятие многоуровневой ЭВМ.~~ Понятие и пример программы на разных уровнях.

\* Структура и принципы работы арифметико-логического устройства. Регистр состояния БЭВМ

\* ~~Архитектура ЭВМ~~. ***~~Гарвардская и фон-Неймановская архитектура~~***~~.~~ ~~Организация обмена архитектуры ЭВМ с использованием шин.~~

\* Архитектура многопроцессорных ЭВМ. Системный коммутатор. Архитектуры UMA и NUMA.

\* ***Структура современных процессоров. Окружение процессора. CISC, RISC, VLIW.***

\* ***Адресуемая память, организация и временные диаграммы. Конструктивные особенности современной памяти.***

\* *~~Характеристики запоминающих устройств. Пирамида памяти.~~*

\* Параллельная передача данных. Контроллеры параллельной передачи и приема.

\* Синхронные последовательные интерфейсы. Контроллеры последовательной передачи и приема.

\* Асинхронный обмен. Принципы деления частоты, формат кадра.

\* Контроллер передачи асинхронного последовательные интерфейса.

\* Контроллер приема асинхронного последовательные интерфейса.

HARD

\* Микропрограммный уровень БЭВМ. Структура МПУ. Форматы горизонтальных и вертикальных

микрокоманд.

\* Микропрограммное управление вентильными схемами. Схема управления. Интерпретатор БЭВМ.

\* **~~Память, ориентированная на записи (блочная память).~~** **~~Организация дисковой памяти и памяти на магнитных лентах~~**.

\* ~~Ассоциативная память, Кэш-память. Влияние промахов кэш-памяти на производительность.~~

\* ~~Интерфейсы ввода-вывода. Контроллеры внешних устройств~~. Уровни стандартизации~~, сопряжения с системной шиной~~, ~~циклы обмена~~. ~~Регистры контроллера.~~

\* ~~Организация прерываний в ЭВМ. Вектора прерываний, контроллер прерывания.~~

\* ~~Организация прямого доступа к памяти~~. ~~Контроллер ПДП~~.

OVERKILL

\* ~~Предназначение и организация виртуальной памяти~~. ~~Сегментно-страничная организация.~~ ~~Устройство управления памятью (MMU),~~ ~~буфер трансляции (TLB).~~

\* *~~Сетевые технологии, Понятие сети ЭВМ, классификация компьютерных сетей. Сообщение и пакет~~.~~Модель взаимодействия открытых систем.~~*

\* *Модель TCP/IP: ~~передающая среда~~, ~~канальный и сетевой уровень~~. ~~Адресация~~, ~~передача и~~*

*~~маршрутизация пакетов.~~*

*\* Модель TCP/IP: ~~выделение адресов (DHCP),~~ ~~сервисы имен~~, ~~транспортный и прикладной уровни.~~*

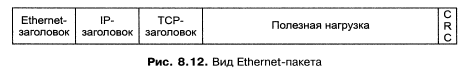
# TCP и прочая хрень

**TCP (Transmission Control Protocol)** – протокол управления передачей

**HTTP (HyperText Transport Protocol)** – протокол передачи гипертекста

**IP (Internet Protocol)** – межсетевой протокол

**CRC (Cyclic Redundancy Check)** – циклический контроль избыточности



**Прикладной уровень (Application Layer)**

Прикладной уровень иерархии протоколов содержит сетевые процессы, с которыми работают пользователи. В тексте книги приложение TCP/IP - это любой сетевой процесс, протекающий выше транспортного уровня. Под это определение подпадают все процессы, с которыми напрямую взаимодействуют пользователи, а также прочие процессы данного уровня, о которых пользователи могут и не знать.

**Транспортный уровень ( Transport Layer)**

Большая часть информации в рассказе о TCP/IP связана с протоколами транспортного уровня. В опорной модели OSI транспортный уровень гарантирует получение адресатом данных в неизмененном виде. В TCP/IP такая функциональность возложена на протокол управления передачей (Transmission Control Protocol, TCP). При этом в TCP/IP существует вторая служба транспортного уровня - протокол пользовательских дейтаграмм (User Datagram Protocol, UDP), который не гарантирует надежной доставки данных.

**Сетевой уровень (Network Layer)**

Сетевой уровень управляет сетевыми соединениями и проводит границу между протоколами более высокого уровня и подробностями реализации собственно сети. Сетевой уровень TCP/IP обычно подразумевает протокол Internet (IP), который разграничивает вышележащие уровни и сеть, а также отвечает за адресацию и доставку данных.

**Канальный уровень (Data Link Layer)**

Надежная доставка данных по физической сети находится в ведении канального уровня. Этот уровень TCP/IP, как правило, не содержит протоколов. В большинстве документов RFC, упоминающих канальный уровень, рассматриваются вопросы интеграции IP и существующих канальных протоколов.

**Пакет** - это блок данных, содержащий информацию, необходимую для доставки

Формат пакета, определяемый протоколом Internet, называется **дейтаграммой**

Из Таненбаума:

**Сетевой уровень** занимается управлением операциями подсети. Важнейшим моментом здесь является определение маршрутов пересылки пакетов от источника к пункту назначения.

Если в подсети одновременно присутствует слишком большое количество пакетов, то они могут закрыть дорогу друг другу, образуя заторы в узких местах. Недопущение подобной закупорки также является задачей сетевого уровня.

При путешествии пакета из одной сети в другую также может возникнуть ряд проблем. Так, способ адресации, применяемый в одной сети, может отличаться от принятого в другой. Также могут различаться протоколы и т.д. Именно сетевой уровень должен разрешить все эти проблемы, позволяя объединять разнородные сети.

Основная функция **транспортного уровня** – принять данные от сеансового уровня, разбить их при необходимости на небольшие части, передать их сетевому уровню и гарантировать, что эти части в правильном виде прибудут по назначению.

**Сеансовый уровень** позволяет пользователям различных компьютеров устанавливать сеансы связи друг с другом.

**Прикладной уровень** содержит набор популярных протоколов, необходимых пользователям. Одним из наиболее распространенных является протокол передачи гипертекста, который составляет основу всемирной паутины. Другие прикладные протоколы используются для передачи файлов, электронной почты, сетевых рассылок.

Протокол динамической настройки узлов **(Dynamic Host Configuration Protocol, DHCP)**

Динамическое выделение Сервер назначает адрес клиенту DHCP на ограниченный период времени. Выделение адреса с ограниченным временем жизни называется арендой (lease). Клиент может вернуть адрес серверу в любой момент, но обязан запросить продление аренды у сервера, чтобы сохранить его дольше предписанного времени жизни. Если клиент не запросил продление, сервер автоматически отбирает адрес. Динамическое выделение адресов - наиболее эффективное из применений DHCP.

**Domain Name System (DNS, Система доменных имен)** Известное также в качестве службы имен, это приложение ассоциирует IP-адреса с именами, назначаемыми сетевым устройствам.

Система доменных имен (DNS) является одной из фундаментальных технологий современной интернет-среды и представляет собой распределенную систему хранения и обработки информации о доменных зонах. Она необходима, в первую очередь, для соотнесения IP-адресов устройств в сети и более удобных для человеческого восприятия символьных имен.

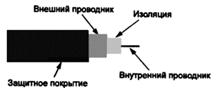
**Физическая передающая среда ЛВС** представлена тремя типами кабелей: витая пара проводов, коаксиальный кабель, оптоволоконный кабель.

***Витая пара*** состоит из двух изолированных проводов, свитых между собой. Скручивание проводов уменьшает влияние внешних электромагнитных полей на передаваемые сигналы. Самый простой вариант витой пары — телефонный кабель.

http://ok-t.ru/studopediaru/baza3/43581940561.files/image194.jpg

Достоинство витой пары – дешевизна. Недостаток витой пары – плохая помехозащищенность и низкая скорость передачи информации – 0,25-1 Мбит/с.

***Коаксиальный кабель*** обладает более высокой механической прочностью, помехозащищённостью и обеспечивает скорость передачи информации до 10-50 Мбит/с. Для промышленного использования выпускаются два вида коаксиальных кабелей: толстый (»10 мм) и тонкий (»4 мм). Толстый кабель более прочен и передает сигналы нужной амплитуды на большее расстояние, чем тонкий. В то же время тонкий кабель значительно дешевле.



***Оптоволоконный кабель*** – идеальная передающая среда. Он не подвержен действию электромагнитных полей и сам практически не имеет излучения. Последнее свойство позволяет использовать его в сетях, требующих повышенной секретности информации.

Для описания структуры и функциональности протоколов обмена данными часто используется архитектурная модель, разработанная Международной организацией по стандартизации (International Standards Organization, ISO). Данная модель как раз является общей точкой отсчета для разговора о коммуникациях и называется **опорной моделью взаимодействия открытых систем (Open Systems Interconnect (OSI) Reference Model).**

|  |  |
| --- | --- |
| **Уровень** | |
| 7 | Прикладной |
| 6 | Представительный |
| 5 | Сеансовый |
| 4 | Транспортный |
| 3 | Сетевой |
| 2 | Канальный |
| 1 | Физический |

Классификации сетей***:***

В зависимости от территориального расположения абонентов компьютерные сети делятся на:

* **глобальные** — вычислительная сеть объединяет абонентов, расположенных в различных странах, на различных континентах. Глобальные вычислительные сети позволяют решить проблему объединения информационных ресурсов человечества и организации доступа к этим ресурсам;
* **региональные** — вычислительная сеть связывает абонентов, расположенных на значительном расстоянии друг от друга. Она может включать абонентов большого города, экономического региона, отдельной страны;
* **локальные** — вычислительная сеть объединяет абонентов, расположенных в пределах небольшой территории. К классу локальных сетей относятся сети отдельных предприятий, фирм, офисов и т. д.

**Коммутация сообщений** по своим принципам близка к коммутации пакетов. Под коммутацией сообщений понимается передача единого блока данных между транзитными компьютерами сети с временной буферизацией этого блока на диске каждого компьютера. Сообщение в отличие от пакета имеет произвольную длину, которая определяется не технологическими соображениями, а содержанием информации, составляющей сообщение. Транзитные компьютеры могут соединяться между собой как сетью с коммутацией пакетов, так и сетью с коммутацией каналов. Сообщение (это может быть, например, текстовый документ, файл с кодом программы, электронное письмо) хранится в транзитном компьютере на диске, причем довольно продолжительное время, если компьютер занят другой работой или сеть временно перегружена.

По такой схеме обычно передаются сообщения, не требующие немедленного ответа, чаще всего сообщения электронной почты. Режим передачи с промежуточным хранением на диске называется режимом **"хранения-и-передачи" (store-and-forward)**

Пакеты обычно тоже могут иметь переменную длину, но в узких пределах, например от 46 до 1500 байт. Каждый пакет снабжается заголовком, в котором указывается адресная информация, необходимая для доставки пакета на узел назначения, а также номер пакета, который будет использоваться узлом назначения для сборки сообщения (рис. 3). Пакеты транспортируются по сети как независимые информационные блоки. Коммутаторы сети принимают пакеты от конечных узлов и на основании адресной информации передают их друг другу, а в конечном итоге — узлу назначения.

# Виртуальная память

В первых компьютерах память была очень мала по объему и дорого стоила. Например, IBM-650 содержал всего 2000 слов памяти.

Традиционным решением проблемы было использование вспомогательной памяти (например, диска). Программист делил программу на несколько частей, так называемых оверлеев, каждый из которых помещался в память. Чтобы выполнить программу, сначала нужно было загрузить первый оверлей, а когда он завершался второй и так далее. Программист отвечал за разбиение программы на оверлеи и решал, в каком месте вспомогательной памяти должен храниться каждый оверлей.

В 1961 году этот процесс автоматизировали исследователи из Манчестера. Этот метод, в основе которого, как сейчас говорят, лежит использование виртуальной памяти.

# Страничная организация памяти

Пусть, у нас будет типичный старый компьютер с 16-разрядным полем адреса и 4096 словами памяти. Программа, работающая на таком компьютере, могла обращаться к 64536 словам памяти. **Адресное пространство** состояло из 0, 1, 2 … 65535. Однако в действительности компьютер мог иметь гораздо меньше памяти.

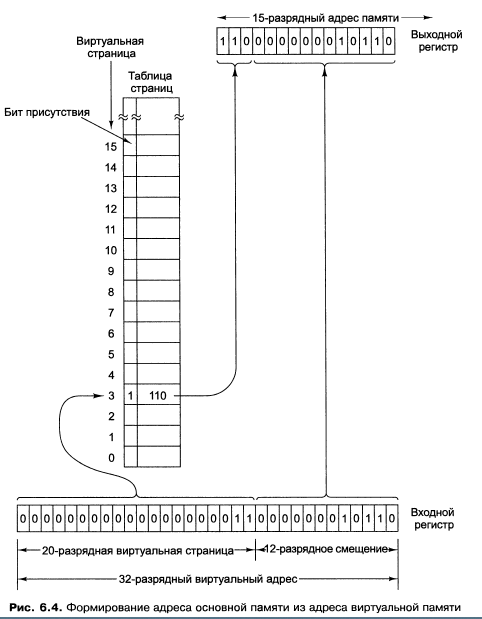
Вводились понятия полезное и бесполезное адресное пространство.

Также можно сообщить компьютеру, что 4096 – это адрес 0, а 4097 – это адрес 1. Но что произойдет, если мы обратимся к 8192 ячейке?

Без виртуальной памяти произойдет ошибка, а если она есть:

* Содержимое основной памяти будет сохранено на диске.
* Слова с 8192 по 12287 будут сохранены на диске.
* Слова с 8192 по 12287 будут загружены в основную память.
* Отображение адресов изменится: адреса с 8192 по 12287 будут соответствовать ячейкам памяти с 0 по 4095
* Выполнение программы продолжится  
    
  Такая технология называется **страничной организацией памяти**, а фрагмента программы, которые считывают с диска, - **страницами**.

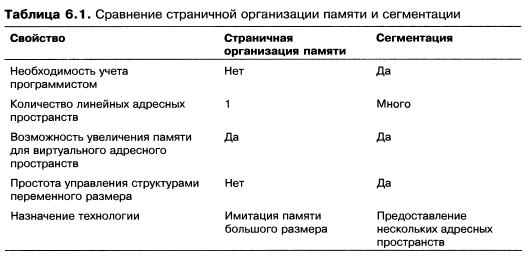




А теперь посмотрим, как 32-разрядный виртуальный адрес можно отобразить

на физический адрес основной памяти. В конце концов, память воспринимает

только реальные, а не виртуальные адреса, поэтому такое отображение должно быть сделано. Каждый компьютер с виртуальной памятью содержит устройство для отображения виртуальных адресов на физические. Это устройство называется **диспетчером памяти (Memory Management Unit, MMU)**. Он может находиться на микросхеме процессора или на отдельной микросхеме рядом с процессором.



Ничто не замедляет работу программы сильнее, чем “узкие места” при работе с памятью. Каждое обращение к памяти требует двух дополнительных обращений к памяти, необходимые для преобразования адреса. 200% дополнительных затрат на каждое обращение к памяти – сильно затормаживают работу программы. Для устранения этого фактора OMAP4430 использует так называемый **буфер быстрого преобразования (Translation Lookaside Buffer, TLB)** для быстрого преобразования номеров виртуальных страниц и номеров физических страничных кадров.

В **TLB** хранятся номера только тех страниц, которые использовались последними. Для каждой из этих категорий в TLB включены номера 128 последних виртуальных страниц. Каждый элемент TLB включает номер виртуальной страницы и соответствующий ему номер физического страничного кадра. Когда номер процесса, называемый ASID, и виртуальный адрес передаются диспетчеру памяти, диспетчер с помощью специальной схемы одновременно сравнивает номер виртуальной страницы сразу со всеми элементами буфера TLB для данного контекста.

* Кэширует часто используемые преобразования
* Обычно раздельный для адреса и данных
* Организован в виде ассоциативной памяти

# ПДП

Необходимость реализации в микроЭВМ обмена данными в режиме ПДП вызывается двумя основными факторами:

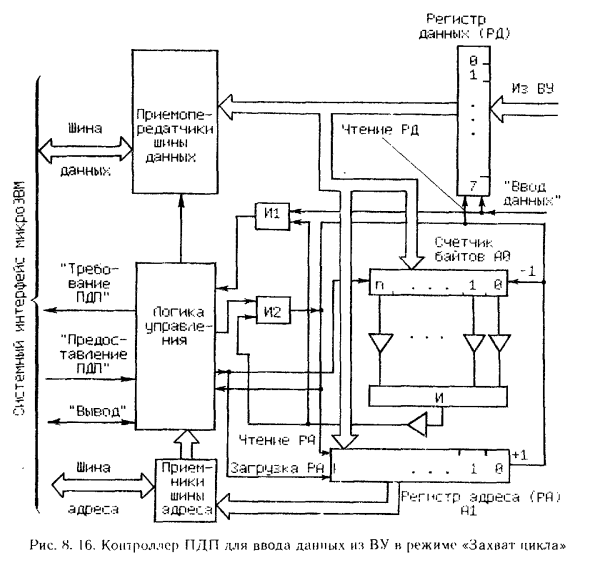
* Появляется возможность начальной загрузки программ
* Обеспечивается возможность использования быстродействующих ВЗУ, таких как накопители на магнитных лентах (НМЛ) и магнитных дисках (НМД)

Для организации такого вида обмена можно выделить специальные шины адреса и данных, но такое решение нельзя признать оптимальным, так как это приведет к значительному усложнению микроЭВМ в целом, особенно при подключении нескольких ВЗУ.

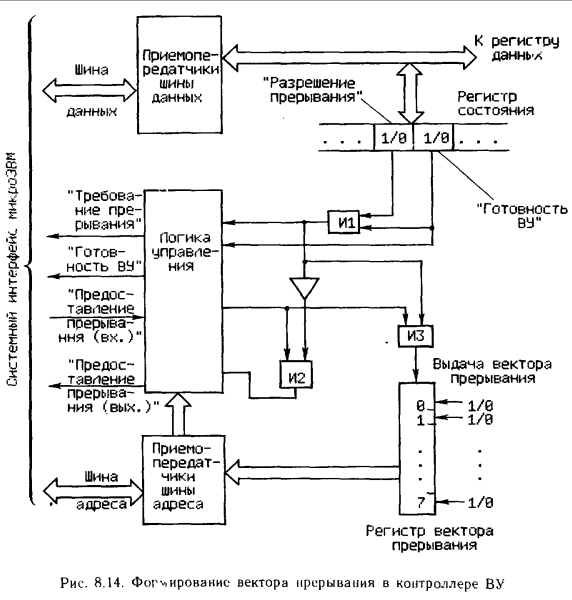
В целях сокращения количества линий в шинах контроллер ПДП подключается к памяти посредством шин адреса и данных системного интерфейса.

Есть два основных способа: реализация обмена в режиме ПДП с “захватом цикла” и в режиме ПДП с блокировкой процессора.

* Наиболее простой способ организации состоит в том, что для обмена используются те машинные циклы процессора, в которых он не обменивается данными с памятью.
* Наиболее распространенным является ПДП с “захватом цикла” и принудительным отключением процессора от шин системного интерфейса. Для реализации такого режима системный интерфейс дополняется двумя линиями для передачи управляющих сигналов “ТПДП” и “ППДП”.  
  Важно заметить, что процессор останавливает свою работу, не дожидаясь окончания выполнения команды.  
  При завершении контроллер снимает ТПДП.



# Вектора прерываний



# Интерфейсы ввода-вывода

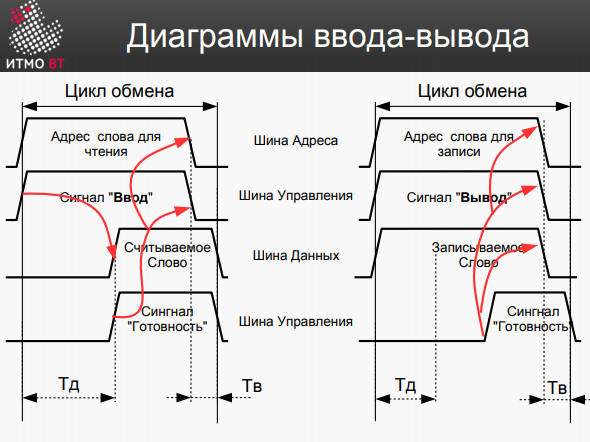
# C:\Users\Артур\YandexDisk-arthur.kupriyanov\Скриншоты\2019-06-16_14-14-08.pngа

* Определяет конкретные детали обмена
* Частота, набор каналов передачи, способ кодирования, команды, представления данных, набор данных и последовательность, …..
* Аппаратная и/или программная реализация
* Нуждаются в точной спецификации и/или стандартизации
* Стороны обмена должны однозначно интерпретировать детали обмена

# Уровни стандартизации интерфейсов

* Логическое подключение
* Физические параметры сигналов
* Конструктивные особенности

# Цикл обмена



# Регистры контроллера

Регистры контроллера:

* Входной
* Выходной
* Состояния
* Управления

Основу контроллера ВУ составляют несколько регистров, которые служат для временного хранения передаваемой информации. Каждый регистр имеет свой адрес, и зачастую такие регистры называют портами ввода-вывода. Регистры входных и выходных данных работают соответственно только в режиме чтения и только в режиме записи. Регистр состояния работает только в режиме чтения и содержит информацию о текущем состоянии ВУ (вкл\выкл, готово\нет …) Регистр управления работает только в режиме записи и служит для приема из микроЭВМ приказов для ВУ. Иногда удается совместить регистр состояния и управления в один регистр, что позволяет сократить количество используемых портов ввода-вывода, а следовательно и адресов выделенных для данного ВУ.

# Кэш-память

Основная идея кэш-памяти проста: в ней находятся слова, которые чаще всего используются. Если процессору нужно какое-нибудь слово, сначала он обращается к кэш-памяти. Только в том случае, если слова там нет, он обращается к основной памяти.

Принцип локальности. Этот принцип составляет основу всех систем кэш-памяти. Идея состоит в том, что когда определенное слово вызывается из памяти, оно вместе с соседними словами переносится в кэш-память, что позволяет при очередном запросе быстро обращаться к следующим словам.

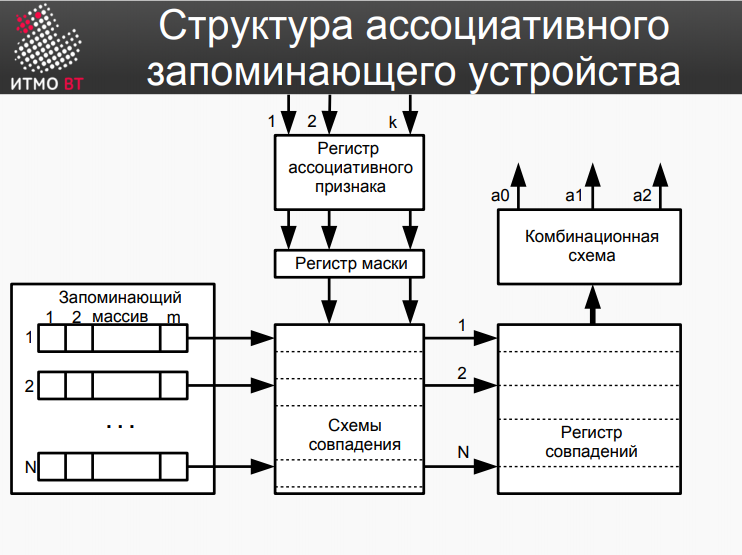
Пусть, с – время доступа к кэш-памяти, m – время доступа к основной памяти и h – коэффициент кэш-попаданий, который показывает соотношение числа обращений к кэш памяти и общего числа обращений.

Тогда, среднее время доступа:

,

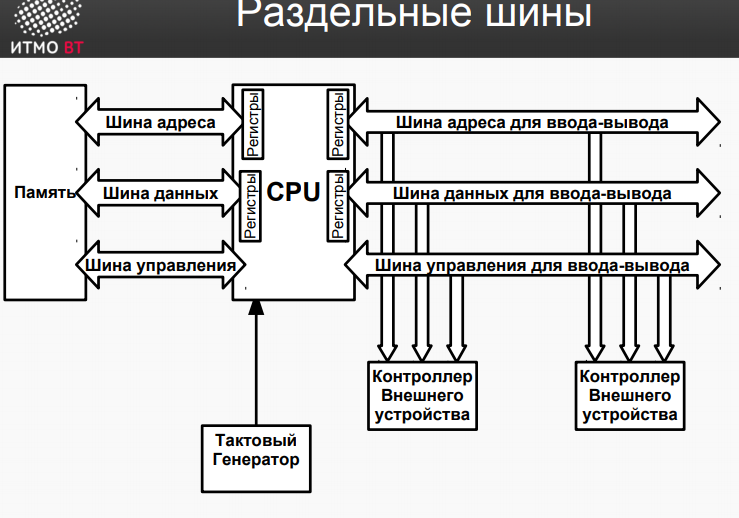
где (1 -h ) – коэффициент кэш-промахов

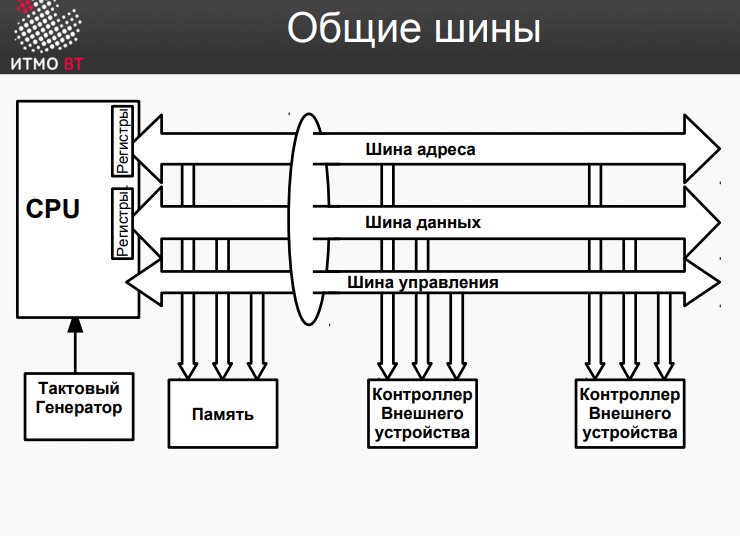
Если h -> 1, то время доступа стремится к “c”, а если h->0, то к (c + m)

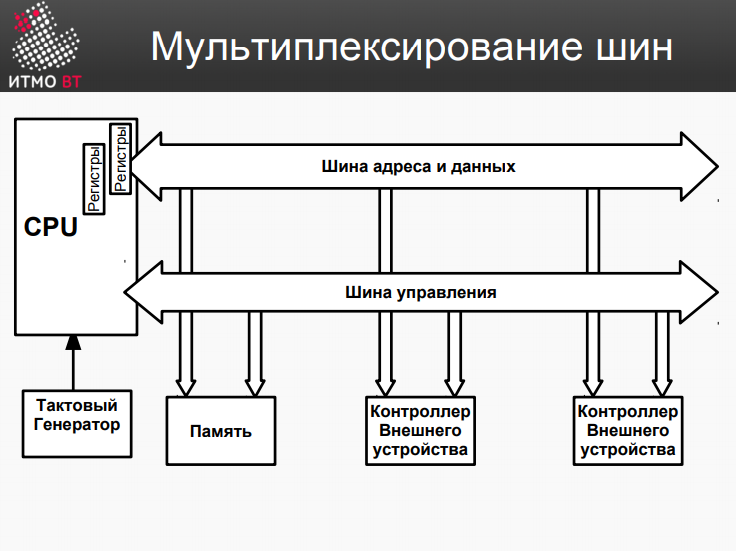


# Организация шин и канальная организация

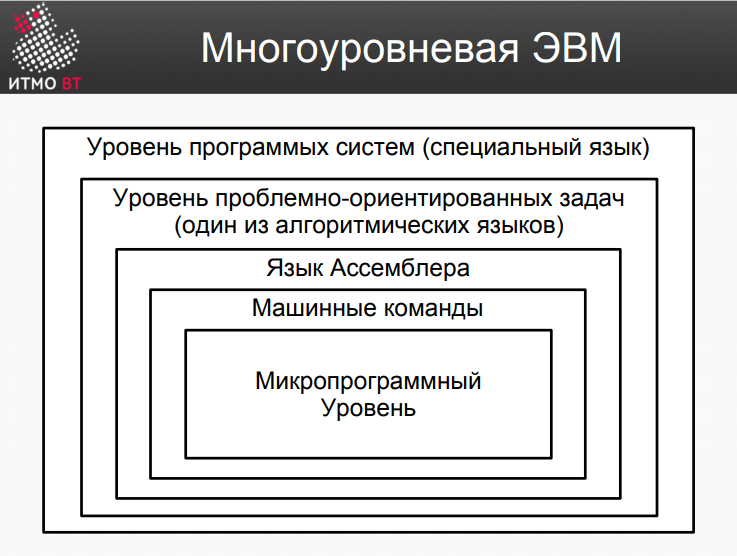








# Уровни ЭВМ



# Функциональные элементы ЭВМ

* Логические элементы
* Элементы хранения SRAM/DRAM
* Элементы хранения триггеры, регистры
* Провода, шины
* Вентили
* Сумматоры (входят в АЛУ)
* Тактовые генераторы