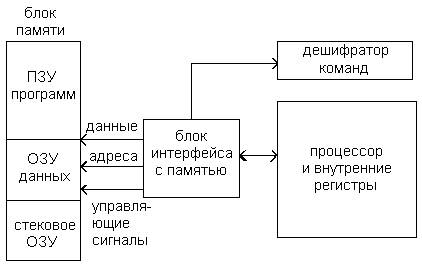
ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 4

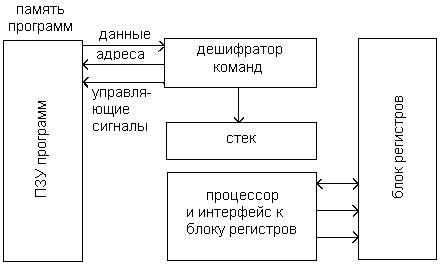
по предмету «Элементы управления в АСОИУ»

1. Принстонская и гарвардская архитектуры: достоинства и недостатки.
2. Микропроцессоры IA-32: организация логического адресного пространства.

**1.Принстонская и гарвардская архитектуры: достоинства и недостатки.**



Структура компьютера с Принстонской архитектурой



Структура компьютера с Гарвардской архитектурой

*Принстонская архитектура* выиграла соревнование, так как она больше соответствовала уровню технологии того времени. Использование обшей па­мяти оказалось более предпочтительным из-за ненадежности ламповой элек­троники (это было до широкого распространения транзисторов) — при этом возникало меньше отказов.

*Гарвардская архитектура* почти не использовалась до конца 70-х годов, когда производители микроконтроллеров поняли, что эта архитектура дает преимущества устройствам, которые они разрабатывали.

Основным преимуществом *архитектуры Фон Неймана (Пристонская\_)* является то, что она упрощает устройство микропроцессора, так как реализует обращение только к одной обшей памяти. Для микропроцессоров самым важным являет­ся то, что содержимое ОЗУ (RAM — Random Access Memory) может быть использовано как для хранения данных, так и для хранения программ. В не­которых приложениях программе необходимо иметь доступ к содержимому стека. Все это предоставляет большую гибкость для разработчика программ­ного обеспечения, прежде всего в области операционных систем реального времени.

*Гарвардская архитектура* выполняет команды за меньшее количество так­тов, чем архитектура Фон Неймана. Это обусловлено тем, что в Гарвардской архитектуре больше возможностей для реализации параллельных операций. Выборка следующей команды может происходить одновременно с выполне­нием предыдущей команды, и нет необходимости останавливать процессор на время выборки команды.

**Концептуально отличие**

Главное отличие этих двух концепций заключается в том, что [архитектура фон Неймана](https://yandex.ru/turbo/fb.ru/s/article/239028/arhitektura-fon-neymana-istoriya-vozniknoveniya-termina?parent-reqid=1609947880595883-332936441680882965700107-production-app-host-vla-web-yp-328&utm_source=turbo_turbo) (принстонская) использовала единую память, т. е. общую шину данных. Что касается ее «конкурентки»,автором которой стал Говард Эйкен, для её реализации требуется наличие нескольких шин.

Кроме того, гарвардская [архитектура вычислительной системы](https://yandex.ru/turbo/fb.ru/s/article/276745/arhitektura-kompyuternoy-sistemyi-klassifikatsiya-i-opredelenie?parent-reqid=1609947880595883-332936441680882965700107-production-app-host-vla-web-yp-328&utm_source=turbo_turbo) отличается от принстонской тем, что ее реализация при достаточной сложности схемы является более скоростной. Это связно с тем, что в фон-неймановском варианте памяти программ и данных не могут быть доступны одновременно.

## Гарвардская архитектура vs принстонская

Как известно, главными составляющими ЭВМ являются АЛУ и память. Очевидно, что чем меньше проводников между ними, тем лучше. С этой точки зрения и с учетом технических возможностей, которые имели место до конца 60-х годов прошлого века, безусловно, лидировала неймановская архитектура. Именно она легла в основу конструкции процессоров RISC.

Но научно-технический прогресс не стоял на месте, и в 70-х годах ХХ века появились полупроводники. С их помощью можно было создавать множество микропроводников, что ликвидировало проблему использования большого количества контактов и привело к наступлению эры гарвардской архитектуры.

## Дальнейшее противостояние

Появление процессоров, разработанных на основе гарвардской архитектуры, было встречено без особого энтузиазма, так как на тот момент не существовало ПО, способного сделать ощутимыми преимущества их использования. В частности, из-за того, что они не могли работать на больших частотах, их называли процессорами для бедных.

Ситуация с востребованностью гарвардской архитектуры изменилась после появления ПК Apple I. Он функционировал на 8-разрядном процессоре MOS 6502, действующем на гарвардской архитектуре и с ОС Apple DOS.

Простота операционной системы компенсировалась сложной конструкцией процессора, названного CISC. Он обладал отдельной 16-разрядной адресной шиной и позволял произвольно манипулировать регистрами. Процессор CISC обладал производительностью, которая в несколько раз превышала все уже существующие.

Вслед за этим IBM повторила идею Apple, создав персональный [компьютер IBM-PC](https://yandex.ru/turbo/fb.ru/s/article/196203/ibm-kompyuteryi-personalnyiy-kompyuter?parent-reqid=1609947880595883-332936441680882965700107-production-app-host-vla-web-yp-328&utm_source=turbo_turbo) с процессором от Intel, функционирующим согласно концепции гарвардской архитектуры. В качестве ОС использовался продукт компании Microsoft — Microsoft DOS. Системы с таким составом называются Wintel.

## Недостатки гарвардской архитектуры ЭВМ

За быстродействие CISC-процессора пришлось заплатить удвоенным/утроенным количеством контактов. Это не только стало причиной его перегрева, но и наложило ограничения на его размеры. В среднем на каждые 20 процентов прироста производительности гарвардского процессора его потребляемая мощность увеличивалась до 50 процентов.

Для решения этой проблемы были изобретены многоядерные процессоры, в которых частота работы каждого отдельного ядра была понижена, но по суммарной производительности они превышали показатели даже разогнанного одноядерного.

## Современная архитектура ПК

Сегодня существуют ЭВМ с разными типами и даже гибридными архитектурами. Однако основными принципами, которые их определяют, являются:

* Программное управление. Оно позволяет автоматизировать процесс вычислений на ЭВМ. Согласно этому принципу, решение любой задачи осуществляется по программе, определяющей последовательность действий ПК.
* Принцип программы, сохраняемой в памяти. Он содержит требование подачи команд в виде чисел, как это происходит в отношении данных, и их обработку таким же способом, что и чисел. При этом сама она перед запуском загружается в оперативную память, что дает возможность ускорить процесс выполнения.
* Принцип произвольного доступа к компьютерной оперативной памяти. Элементы программ и данных записываются в любое место ОП. Такое решение позволяет обратиться к конкретному участку памяти, не просматривая предыдущие.
* Микропроцессоры IA-32: организация логического адресного пространства.

**2.Микропроцессоры IA-32: организация логического адресного пространства.**

Архитектура компьютера различает физическое адресное пространство (ФАП) и логическое адресное пространство (ЛАП).

**Физическое адресное пространство** представляет собой простой одномерный массив байтов, доступ к которому реализуется аппаратурой памяти по адресу, присутствующему на шине адреса микропроцессорной системы.

**Логическое адресное пространство** организуется самим программистом исходя из конкретных потребностей. Трансляцию логических адресов в физические осуществляет блок управления памятью MMU.

В архитектуре современных микропроцессоров ЛАП представляется в виде набора элементарных структур: байтов, сегментов и страниц.

В микропроцессорах используются следующие варианты организации **логического адресного пространства**:

* **плоское (линейное) ЛАП**: состоит из массива байтов, не имеющего определенной структуры; трансляция адреса не требуется, так как логический адрес совпадает с физическим;
* **сегментированное ЛАП**: состоит из сегментов - непрерывных областей памяти, содержащих в общем случае переменное число байтов; логический адрес содержит 2 части: идентификатор сегмента и смещение внутри сегмента; трансляцию адреса проводит блок сегментации MMU;
* **страничное ЛАП**: состоит из страниц - непрерывных областей памяти, каждая из которых содержит фиксированное число байтов. Логический адрес состоит из номера (идентификатора) страницы и смещения внутри страницы; трансляция логического адреса в физический проводится блоком страничного преобразования MMU;
* **сегментно-страничное ЛАП**: состоит из **сегментов**, которые, в свою очередь, состоят из страниц; логический адрес состоит из идентификатора сегмента и смещения внутри сегмента. Блок сегментного преобразования MMU проводит трансляцию логического адреса в номер страницы и смещение в ней, которые затем транслируются в физический адрес блоком страничного преобразования MMU.

Таким образом, *основой получения физического адреса памяти служит логический адрес.* В какой-то степени логическое адресное пространство, с которым имеет дело программист, можно сравнить со структурой книги, где аналогом сегмента выступает рассказ, страница книги соответствует странице ЛАП, а искомая информация - это некоторое слово. При этом если память организована как ***линейная,*** то номер искомого слова задается в явном виде и просто отсчитывается от начала книги.

При ***сегментном*** представлении памяти искомое слово определяется его номером в заданном рассказе.

***Страничное*** представление памяти предполагает задание информации о слове в виде номера страницы в книге и номера слова на указанной странице.

При ***сегментно-страничном*** представлении логический адрес слова задается номером слова в определенном рассказе. В этом случае по оглавлению книги определяется номер страницы, с которой начинается указанный рассказ. Затем, зная количество слов на странице и положение слова в рассказе, можно вычислить страницу книги и положение искомого слова на этой странице.