


## **ЛЕКЦИЯ 9-10**

**Технологии повышения производительности процессоров  
Основные принципы работы современных процессоров Pentium  
Концепция многоуровневой памяти**

## Конвейерная обработка команд

*Конвейеризация* (pipelining) осуществляет многопоточную параллельную обработку команд, так что, в каждый момент времени одна из команд считывается, другая декодируется и т. д., всего в обработке находится пять команд.

При выполнении инструкция продвигается по конвейеру по мере освобождения последующих ступеней. Таким образом, на конвейере одновременно может обрабатываться несколько последовательных инструкций, и производительность процессора можно оценивать темпом выхода выполненных инструкций со всех его конвейеров. Такая технология обработки команд носит название *конвейерной* (pipeline) обработки. Каждая ступень устройства называется *ступенью конвейера*, а общее число ступеней – *длиной конвейера*.




*Скалярным* называют процессор с единственным конвейером, к этому типу относятся процессоры Intel до 486 включительно. Суперскалярный (superscalar) процессор имеет более одного конвейера, способных обрабатывать инструкции параллельно.

Pentium – первый суперскалярный процессор Intel является двухпоточковым процессором (имеет два конвейера, т.е. при одинаковых частотах вдвое производительней i80486, выполняя сразу две инструкции за такт), PentiumPro – трехпоточковым.


Наиболее высокопроизводительной ВС является GRAY – максимальная пиковая производительность процессора может составлять 12 Гфлопс. (FloatingPointOperationsperSecond – FLOPS – мера быстродействия в операциях с плавающей точкой за секунду.)

## Характеристика конвейеров процессоров Intel

Процессор	I80486	Pentium	Pentium-Pro	Pentium-MMX	PentiumII	PentiumIII	PentiumIV
Число линий	1	2	3	2	3	3	3
Длина линии	5	5	14	6	14	20	31 (HyperPipe-line)



*Технология IntelMMX* улучшает компрессию/декомпрессию видео, работу с изображениями, шифрование и обработку сигналов ввода/вывода - т.е. все мультимедиа-операции, операции связи и сетевые взаимодействия. Основа MMX расширения процессорного ядра заключается в *технологии обработки множественных данных в одной инструкции (SingleInstructionMultipleData - SIMD)*. Процесс SIMD (один поток команд и множество потоков данных) дает возможность одной инструкции исполнять одну и ту же функцию с различными данными и их частями. SIMD позволяет чипу уменьшить количество циклов с интенсивными вычислениями, характерными для обработки видео, аудио, графической информации и анимации.



**Матричные и векторные процессоры.** Матричный процессор имеет архитектуру, рассчитанную на обработку числовых массивов. Архитектура процессора включает в себя матрицу процессорных элементов, например 64x64, работающих одновременно.

Векторный процессор обеспечивает параллельное выполнение операции над массивами данных, векторами. Он характеризуется специальной архитектурой, построенной на группе параллельно работающих процессорных элементов.

Векторная обработка увеличивает производительность процессора за счет того, что обработка целого набора данных (вектора) производится одной командой. Максимальная скорость передачи данных в векторном формате может составлять 64 Гбайт/с. В настоящее время созданы однокристалльные векторно-конвейерные процессоры, такие как SX-6. Основные компоненты - скалярный процессор и восемь идентичных векторных устройств, суммарная производительность которых составляет 64 Гфлопс. Например, процессоры фирм NEC и Hitachi.

**Динамическое исполнение (Dynamic execution technology)** это совокупность технологий обработки данных в процессоре, обеспечивающая более эффективную работу процессора за счет манипулирования данными, а не простого исполнения списка инструкций. Динамическое исполнение представляет собой комбинацию трех методов обработки данных:

- множественное предсказание ветвлений;
- анализ потока данных;
- спекулятивное (по предположению) исполнение.

**4. Hyper-Pipelined Technology (HT).** Здесь реализуется разделение времени на аппаратном уровне: физически процессор разбивается на два логических процессора, каждый из которых использует ресурсы чипа - ядро, кэш-память, шины, исполнительное устройство. Ядро выполняет два процесса одновременно. Специалисты Intel оценивают повышение эффективности в 30% при использовании на HT-процессорах многопрограммных ОС и прикладных программ.

# Семейство процессоров Intel

История 32-разрядных процессоров Intel началась с процессора Intel386. Он вобрал в себя все черты 16-разрядных предшественников 8086/88 и 80286 для обеспечения совместимости с громадным объемом программного обеспечения, существовавшего на момент его появления.

Однако в процессорах 80386 преодолено жесткое ограничение на длину непрерывного сегмента памяти – 64 Кбайт.



## Основные принципы работы современных процессоров Pentium

Pentium 4, одноядерный x86-совместимый микропроцессор компании Intel, был представлен 20 ноября 2000г., и стал первым микропроцессором, в основе которого лежала принципиально новая по сравнению с предшественниками архитектура седьмого поколения (по классификации Intel) — NetBurst. К процессорам архитектуры NetBurst относятся двухъядерные процессоры Pentium D, а также некоторые процессоры Xeon, предназначенные для серверов, часть процессоров Celeron, предназначенных для систем нижнего ценового уровня.


Процессор Pentium состоит из следующих блоков:

*Ядро (Core)* – основное исполнительное устройство.

Производительность процессора при тактовой частоте 66 МГц составляет около 112 млн. инструкций в секунду (Mips). Пятикратное повышение (по сравнению с 80486 DX) достигалось благодаря двум 5-ступенчатым конвейерам, позволяющим выполнить одновременно несколько инструкций.

Целочисленные команды могут выполняться за один такт синхронизации. Эти процессоры имеют встроенный блок управления памятью, который поддерживает механизмы сегментации и страничной трансляции адресов (Paging).


*Предсказатель переходов (BranchPredictor), Буфер адреса переходов (BranchTargetBuffer) – для загрузки и предсказания переходов. Блок плавающей точки (FloatingPointUnit) – выполняет обработку чисел с плавающей точкой. Кэш-память 1-го уровня (Level 1 cache, L1). Процессор имеет два банка памяти по 8 Кбайт: 1-й – для команд, 2-й – для данных, которые обладают большим быстродействием, чем более емкая внешняя кэш-память (L2 cache).*



*Интерфейс шины (BusInterface).* Передает в процессор поток команд и данных, а также передает данные из процессора. Процессоры обеспечивают четырехуровневую систему защиты пространств памяти и ввода/вывода, а также переключение задач. Система команд расширена при сохранении всех команд 8086, 80286. В архитектуру процессоров введены средства отладки и тестирования.

## Общие принципы организации памяти ЭВМ.

В основе большинства ЭВМ лежит трехуровневая организация памяти: сверхоперативная (СО-ЗУ, кэш память CacheMemory) – оперативная (ОЗУ) – внешняя (ВЗУ). СОЗУ и ОЗУ могут непосредственно взаимодействовать с процессором, ВЗУ взаимодействует только с ОЗУ. СОЗУ обладает максимальным быстродействием (равному процессорному), небольшим объемом ( $10^1$  –  $10^5$  байтов) и располагается, как правило на кристалле процессорной БИС(инт.схем). Для обращения к СОЗУ не требуется магистральные (машинные) циклы. В СОЗУ размещаются наиболее часто используемые на данном участке программы данные, а иногда – и фрагменты программы. Обычно памятью машины называют оперативное запоминающее устройство ОЗУ (MainMemory). ОЗУ используется для записи программ, а также исходных данных, промежуточных и конечных результатов. Быстродействие ОЗУ может быть ниже процессорного (не более чем на порядок), а объем составляет  $10^6$  –  $10^9$  байтов. В ОЗУ располагаются подлежащие выполнению программы и обрабатываемые данные. Связь между процессором и ОЗУ осуществляется по системному или специализированному интерфейсу и требует для своего осуществления машинных циклов. ОЗУ не сохраняет информацию при отключении питания.



Информация, находящаяся в ВЗУ, не может быть непосредственно использована процессором. Для использования программ и данных, расположенных в ВЗУ, их необходимо предварительно переписать в ОЗУ.

Процесс обмена информацией между ВЗУ и ОЗУ осуществляется средствами специального канала или (реже) – непосредственно под управлением процессора.

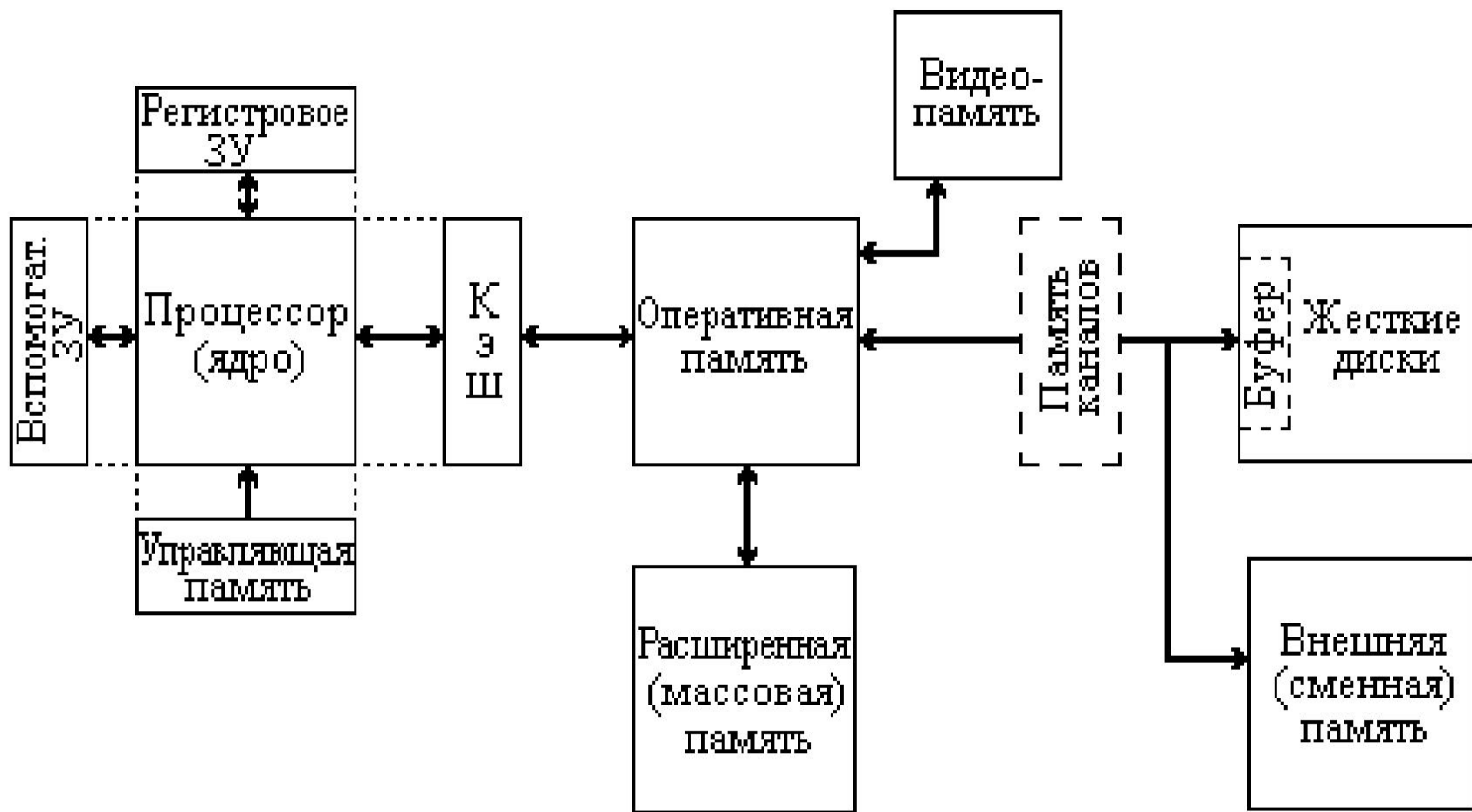
Объем ВЗУ практически неограничен, а быстродействие на 3-6 порядков ниже процессорного.

Положение ЗУ в иерархии памяти ЭВМ определяется не элементной базой запоминающих ячеек, а возможностью доступа процессора к данным, расположенным в этом ЗУ.

При организации памяти особое внимание уделяется сверхоперативной памяти и принципам обмена между ОЗУ и ВЗУ.

## Классификация по функциональному назначению

1 Верхнее место в иерархии памяти занимают *регистровые ЗУ*, которые входят в состав процессора и часто рассматриваются не как самостоятельный блок ЗУ, а просто как набор регистров процессора. Такие ЗУ в большинстве случаев реализованы на том же кристалле, что и процессор, и предназначены для хранения небольшого количества информации (до нескольких десятков слов, а в RISC-архитектурах – до сотни), которая обрабатывается в текущий момент времени или часто используется процессором. Это позволяет сократить время выполнения программы за счет использования команд типа регистр-регистр и уменьшить частоту обменов информацией с более медленными ЗУ ЭВМ. Обращение к этим ЗУ производится непосредственно по командам процессора.

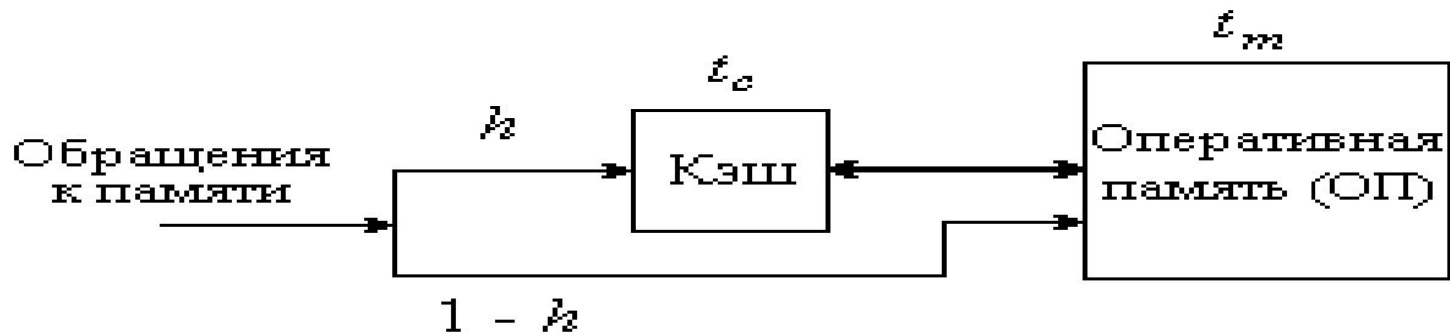


Возможный состав системы памяти ЭВМ

2. Следующую позицию в иерархии занимают *буферные ЗУ*. Их назначение состоит в сокращении времени передачи информации между процессором и более медленными уровнями памяти компьютера. Ранее такие буферные ЗУ в литературе называли сверхоперативными, сейчас это название практически полностью вытеснил термин "кэш-память" или просто *кэш*.

Буфер представляет собой более быстрое (а значит, и более дорогое), но менее емкое ЗУ, чем то, для ускорения работы которого он предназначен. При этом в буфере размещается только та часть информации из более медленного ЗУ, которая используется в настоящий момент. Если доля  $h$ -обращений к памяти со стороны процессора, удовлетворяемых непосредственно буфером (кэшем) высока (0,9 и более), то среднее время для всех обращений оказывается близким ко времени обращения к кэшу, а не к более медленному ЗУ. Пусть двухуровневая память состоит из кэш и оперативной памяти. И пусть, например, время обращения к кэшу  $t_c = 1$  нс ( $10^{-9}$  с), время  $t_m$  обращения к более медленной памяти в десять раз больше –  $t_m = 10$  нс, а доля обращений, удовлетворяемых кэшем,  $h = 0,95$ . Тогда среднее время обращения к такой двухуровневой памяти  $T_{cp}$  составит  $T_{cp} = 1 * 0,95 + 10 * (1 - 0,95) = 1,45$  нс, т.е. всего на 45% больше времени обращения к кэшу. Значение  $h$  зависит от размера кэша и характера выполняемых программ и иногда называется отношением успехов или попаданий (*hitratio*).





К расчету среднего времени обращения

( $t_c$  - время обращения к кэш-памяти,  $t_m$  - время обращения к ОП,  $h$  - доля обращения, обслуживаемых кэш-памятью,  $1 - h$  - доля обращений, обслуживаемых ОП)

3 Еще одним (внутренним) уровнем памяти являются *служебные ЗУ*. Они могут иметь различное назначение. Одним из примеров таких устройств являются ЗУ микропрограмм, которые иногда называют управляющей памятью. Другим - вспомогательные ЗУ, используемые для управления многоуровневой памятью.

4 *Оперативное ЗУ (ОЗУ)* является основным запоминающим устройством ЭВМ, в котором хранятся выполняемые в настоящий момент процессором программы и обрабатываемые данные, резидентные программы, модули операционной системы и т. п. Название оперативной памяти также несколько изменялось во времени. В некоторых семействах ЭВМ ее называли основной памятью, основной оперативной памятью и пр. В англоязычной литературе также используется термин *RAM (randomaccessmemory)*, означающий память с произвольным доступом. Оперативная память реализуется на полупроводниках (интегральных схемах), стандартные объемы ее составляют (в начале 2000-х годов) сотни мегабайт - единицы гигабайт, а времена обращения - единицы÷десятки наносекунд.

5 *Дополнительная память*, которую иногда называли *расширенной или массовой*. Первоначально (1970-е годы) эта ступень использовалась для наращивания емкости оперативной памяти до величины, соответствующей адресному пространству (например, 24-битного адреса) команд, с помощью подключения более дешевого и емкого, чем ОЗУ, запоминающего устройства. Это могла быть ферритовая память или даже память на магнитных дисках. Конечно, она была более медленной, а хранимая в ней информация передавалась в оперативную память и только оттуда попадала в процессор. При записи путь был обратный.

6 В состав памяти ЭВМ входят также ЗУ, принадлежащие отдельным функциональным блокам компьютера. Формально эти устройства непосредственно не обслуживают основные потоки данных и команд, проходящие через процессор. Их назначение обычно сводится к буферизации данных, извлекаемых из каких-либо устройств и поступающих в них.

Типичным примером такой памяти является *видеопамять* графического адаптера, которая используется в качестве буферной памяти для снижения нагрузки на основную память и системную шину процессора.

Другими примерами таких устройств могут служить *буферная память* контроллеров жестких дисков, а также память, использовавшаяся в каналах (процессорах) ввода-вывода для организации одновременной работы нескольких внешних устройств.

Емкости и быстродействие этих видов памяти зависят от конкретного функционального назначения обслуживаемых ими устройств. Для видеопамяти, например, объем может достигать величин, сравнимых с оперативными ЗУ, а быстродействие – даже превосходить быстродействие последних.

7 Следующей ступенью памяти, ставшей фактически стандартом для любых ЭВМ, являются *жесткие диски*. В этих ЗУ хранится практически вся информация, которая используется более или менее активно, начиная от операционной системы и основных прикладных программ и кончая редко используемыми пакетами и справочными данными.

8 Все остальные запоминающие устройства можно объединить с точки зрения функционального назначения в одну общую группу, охарактеризовав ее как группу *внешних ЗУ*. Под словом «внешние» следует подразумевать то, что информация, хранимая в этих ЗУ, в общем случае расположена на носителях не являющихся частью собственно ЭВМ. Под это определение подпадают гибкие диски, компакт диски, накопители на сменных магнитных дисках и магнитооптические диски, твердотельные (флэш) диски и флэш-карты, стримеры, внешние винчестеры и др. Естественно, что параметры этих устройств достаточно различны. Функциональное назначение их обычно сводится либо к архивному хранению информации, либо к переносу ее од одного компьютера к другому.

## Классификация по принципу организации

Особенности организации ЗУ определяются, используемыми технологиями, логикой их функционирования, а также некоторыми другими факторами:

1. По *функциональным возможностям* ЗУ можно разделять: на простые, допускающие только хранение информации; многофункциональные, которые позволяют не только хранить, но и перерабатывать хранимую информацию без участия процессора непосредственно в самих ЗУ. Подход, используемый во второй группе ЗУ, в принципе, позволяет создать производительные системы с параллельной обработкой данных.

2. По *возможности изменения информации* различают ЗУ: постоянные (или с однократной записью); односторонние (с перезаписью или перепрограммируемые); двусторонние.

3. По *способу доступа* различают ЗУ: с адресным доступом; с ассоциативным доступом.

4. По *организации носителя* различают ЗУ: с неподвижным носителем; с подвижным носителем.

5. По *возможности смены носителя* ЗУ могут быть: с постоянным носителем; со сменным носителем.

6. По *способу подключения* к системе ЗУ делятся: на внутренние (стационарные); внешние (съёмные).

7. По *количеству блоков*, образующих модуль или ступень памяти, можно различать: одноблочные ЗУ; многоблочные ЗУ.

## Защита информации и памяти

Для повышения достоверности информации, хранимой на различных уровнях иерархии памяти, применяются *дублирование, избыточное кодирование и ограничение доступа*. На практике часто используют комбинированное сочетание этих способов.

Еще одной формой проверки ошибок служит подсчет *контрольных сумм*. Это несложный способ, который обычно применяется вместе с проверкой на четность/нечетность. Сущность его состоит в суммировании численных значений всех ячеек блока памяти. Шестнадцать младших разрядов суммы помещаются в 16-разрядный счетчик контрольной суммы, который вместе с информацией пользователей записывается в память. При считывании выполняются такие же вычисления и сравнивается полученная контрольная сумма с записанной. Если эти суммы совпадают, подразумевается, что блок без ошибок. При этом имеется незначительная вероятность того, что в результате такой проверки ошибочный блок может быть не обнаружен.

Код Хэмминга позволяет не только обнаруживать, но и исправлять ошибки. Код Хэмминга – метод определения и исправления ошибок при передаче данных, использующий проверочные биты и проверочную сумму.

В этом коде каждая кодовая комбинация состоит из  $m$  информационных и  $k$  контрольных элементов, так, например, в семиэлементном коде Хэмминга  $n=7$ ,  $m=4$ ,  $k=3$  (для всех остальных элементов существует специальная таблица). Контрольные символы 0 или 1 записываются в первый, второй и четвертый элементы кодовой комбинации, причем в первый элемент – в соответствии с контролем на четность для третьего, пятого и седьмого элементов, во второй – для третьего, шестого и седьмого элементов, и в четвертый – для пятого, шестого и седьмого элементов. В соответствии с этим правилом комбинация 1001 будет представляться в коде Хэмминга как 0011001, и в этом виде она будет записываться в ячейку памяти.

При декодировании в начале проверяются на четность первый, третий, пятый и седьмой элементы, результат проверки записывается в первый элемент контрольного числа. Далее контролируются четвертый – седьмой элементы – результат проставляется в младшем элементе контрольного числа.

При правильно выполненной передаче контрольное число состоит из одних нулей, а при неправильной – из комбинации нулей и единиц, соответствующей при чтении ее справа налево номеру элемента, содержащего ошибку.

Для устранения этой ошибки необходимо изменить находящийся в этом элементе символ на обратный.

Код Хэмминга имеет существенный недостаток: при обнаружении любого числа ошибок он исправляет лишь одиночные ошибки.

Избыточность семиэлементного кода Хэмминга равна 0,43. При увеличении значности кодовых комбинаций увеличивается число проверок, но уменьшается избыточность кода.

# 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.