

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН**

УНИВЕРСИТЕТ МИРАС

**Игнашова Людмила Викторовна
Жукова Татьяна Алексеевна**

АРХИТЕКТУРА КОМПЬЮТЕРНЫХ СИСТЕМ

Учебное пособие

Шымкент, 2017

УДК 004(075.8)

ББК 32.973 я7

И26

Обсуждена и рассмотрена на Учебно-методическом совете (Протокол № 11 от 23.06.2017 года)

Рецензенты:

Тенизбаев Е.Ж. – к.т.н, доцент кафедры «Вычислительная техника и дизайн» ЮКГИ им. М. Сапарбаева

Джаманкараева М.А. – к.ф.-м.н, старший преподаватель кафедры «Вычислительная техника и дизайн» ЮКГИ им. М. Сапарбаева

Жукова Т.А. – к.т.н, доцент кафедры информационных технологий и телекоммуникаций университета «Мирас»

Игнашова Л.В., Жукова Т.А.

И26 Архитектура компьютерных систем: учебное пособие. – Шымкент, Университет «Мирас», 2017. - 186 с.

ISBN 978-9965-895-48-7

В учебном пособии в доступной форме представлены основные сведения о принципах функциональной и структурной организации вычислительных машин, систем, комплексов и сетей ЭВМ, арифметических, логических и схемотехнических основах ЭВМ; принципах организации внутренних и внешних запоминающих устройств; структуре процессоров; принципах работы устройств ввода и вывода информации и организации взаимодействия их с центральными устройствами; действиях с арифметикой двоичных чисел; основах проектирования вычислительных систем и сетей.

Книга предназначена для преподавателей ВУЗов и колледжей, а также для студентов, обучающихся по специальностям 5В060200 - «Информатика», 5В070400 - «Вычислительная техника и программное обеспечение», 5В070300 - «Информационные системы» в процессе изучения дисциплин «Архитектура компьютерных систем», «Организация и функционирование персонального компьютера», «Микропроцессорная техника».

УДК 004(075.8)

ББК 32.973 я7

ISBN 978-9965-895-48-7

Игнашова Л.В., Жукова Т.А., 2017

Содержание

ВВЕДЕНИЕ	4
Тема 1. Основы организации ЭВМ.....	8
Тема 2. Представление информации в ЭВМ. Арифметические и логические основы ЭВМ.....	21
Тема 3. Функциональные узлы ЭВМ.....	33
Тема 4. Запоминающие устройства (ЗУ) ЭВМ.....	47
Тема 5. Оперативная память (РАМ).....	57
Тема 6. Масочные, программируемые и репрограммируемые постоянные запоминающие устройства (ROM, PROM, EPROM, EEPROM). Флэш-память. Память с последовательным доступом. Видеопамять. Буферная память типа FIFO. Стековая память (LIFO)....	73
Тема 7. Процессоры ЭВМ.....	93
Тема 8. Центральное устройство управления (ЦУУ).....	103
Тема 9. Организация ввода - вывода. Вычислительные комплексы.....	123
Тема 10. Интерфейсы передачи данных.....	130
Тема 11. Распределенные системы обработки данных.....	139
Тема 12. Основы архитектуры персональных компьютеров.....	154
Тема 13. Структура микропроцессора.....	166
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	183

Введение

Целью изучения дисциплины «Архитектура компьютерных систем» является овладение:

- архитектурными особенностями современных ЭВМ и компьютерных систем;
- основами проектирования функциональных узлов и устройств ЭВМ;
- основами организации вычислительных комплексов.

В процессе изучения дисциплины студенты должны получить систематизированные знания об архитектуре компьютерных систем, организации и основных принципах работы устройств ЭВМ, в частности запоминающих устройств, процессоров и вычислительных комплексов в целом.

Основные задачи изучения дисциплины включают формирование у студентов знаний в следующих направлениях:

- принципы функциональной и структурной организации вычислительных машин, систем, комплексов и сетей ЭВМ, арифметических, логических и схемотехнических основ ЭВМ;
- принципы организации внутренних и внешних ЗУ;
- структуру процессоров;
- принципы работы устройств ввода и вывода информации и организация взаимодействия их с центральными устройствами;
- действий с арифметикой двоичных чисел;
- основы проектирования вычислительных систем и сетей.

В процессе изучения дисциплины студенты должны получить систематизированные знания об архитектуре компьютерных систем, организации и основных принципах работы устройств ЭВМ, в частности запоминающих устройств, процессоров и вычислительных комплексов в целом.

Вычислительная техника прошла в своем развитии 4 стадии:

- компьютеры 1 поколения строились на электронных лампах;
- компьютеры 2 поколения изготавливались с использованием транзисторов;
- компьютеры 3 поколения использовали интегральные микросхемы (интегральная схема - это специальный кристалл кремния или германия, на котором сформированы участки со свойствами полупроводниковых электронных элементов (транзисторов, диодов, конденсаторов и т.п.; использование интегральных схем позволило существенно снизить размеры компьютеров, их стоимость и энергопотребление, увеличить быстродействие и объем памяти);
- компьютеры 4 и 5 поколений используют микропроцессор.

Первое поколение (1955-1960 гг.) строилось на дискретных элементах и вакуумных лампах, имели большие габариты, вес, мощность, обладая при этом малой надежностью. Они использовались в основном для решения научно-технических задач атомной промышленности, реактивной авиации и ракетостроения.

Увеличению количества решаемых задач препятствовали низкая надежность и производительность, а также чрезвычайно трудоемкий процесс подготовки, ввода и отладки программы, написанной на языке машинных команд, т.е. в форме двоичных кодов.

Машины этого поколения имели быстродействие порядка 10-20 тысяч операций в секунду и ОП порядка 1К (1024 слова). В этот же период появились первые простые языки для автоматизированного программирования.

Второе поколение (1960-1965 гг.) использовали дискретные полупроводниковые приборы и миниатюрные дискретные детали, а в качестве технологической - печатный монтаж. По сравнению с предыдущим поколением резко уменьшились габариты и энергозатраты, возросла надежность. Возросли также быстродействие (приблизительно 500 тысяч оп/сек) и объем оперативной памяти (16-32К слов). Это сразу расширило круг пользователей, а, следовательно, и решаемых задач. Появились языки высокого уровня (Фортран, Алгол, Кобол) и соответствующие им трансляторы.

Совершенствование аппаратного обеспечения, построенного на полупроводниковой базе, привело к тому, что появилась возможность строить в ЭВМ помимо центрального (основного) процессора еще ряд вспомогательных процессоров. Эти процессоры управляли всей периферией, в частности устройствами ввода/вывода, избавляли от вспомогательной работы центральный процессор. Одновременно совершенствовались и ОС. Это позволило на ЭВМ второго поколения реализовать режим пакетной обработки программ, а также режим разделенного времени, необходимый для параллельного решения нескольких задач управления производством. На машинах второго поколения были впервые опробованы ОП на ферритовых кольцах (так называемые кубы памяти). Все это позволило поднять производство ЭВМ и привлечь к ней массу новых пользователей.

В ЭВМ *третьего поколения* (1965-1970 гг.) в качестве элементной базы использовались интегральные схемы малой интеграции с десятками активных элементов на кристалл, а также гибридные микросхемы из дискретных элементов. Это позволило сократить габариты и мощность, повысить быстродействие, снизить стоимость универсальных (больших) ЭВМ. Но самое главное - появилась возможность создания малогабаритных, надежных, дешевых машин - *мини-ЭВМ*.

Четвертое поколение (с 1970 г.). Успехи микроэлектроники позволили создать БИС и СБИС, содержащие десятки тысяч активных элементов. Это позволило разработать более дешевые ЭВМ с большой ОП. Стоимость одного байта памяти и одной машинной операции резко снизилась. Но затраты на программирование почти не сократились. Поэтому на первый план вышла задача экономии человеческих, а не машинных ресурсов.

Для этого разрабатывались новые ОС, позволяющие пользователю вести диалог с ЭВМ. Это облегчало работу пользователя и ускоряло разработку программ. Это потребовало, в свою очередь, организовать одновременный доступ к ЭВМ целого ряда пользователей, работающих с терминалов.

В 70-е же годы появились первые *микро-ЭВМ* - универсальные вычислительные системы, состоящие из процессора, памяти, схем сопряжения с устройствами ввода/вывода и тактового генератора, размещенные в одной БИС (*однокристалльная ЭВМ*) или в нескольких БИС, установленных на одной печатной плате (*одноплатные ЭВМ*).

Совершенствование технологии позволило изготовить СБИС, содержащие сотни тысяч активных элементов, и сделать их достаточно дешевыми. Это привело к созданию небольшого настольного прибора, в котором размещалась микро-ЭВМ, клавиатура, экран, магнитный накопитель (кассетный или дисковый), а также схема сопряжения с малогабаритным печатающим устройством, измерительной аппаратурой, другими ЭВМ и т.д. Этот прибор получил название *персональный компьютер*.

В 1976г. была зарегистрирована компания Apple Comp (Стив Джекоб и Стефан Возняк), которая и начала выпуск первых в мире персональных компьютеров "Макинтош".

Благодаря ОС, обеспечивающей простоту общения с этой ЭВМ больших библиотечных прикладных программ, а также низкой стоимости, персональный компьютер начал стремительно внедряться в различные сферы человеческой деятельности во всем мире. Об областях и целях его использования можно прочитать в многочисленных литературных источниках. По данным на 1985 год общий объем мирового производства уже составил $200 \cdot 10^6$ микропроцессоров и $10 \cdot 10^6$ персональных компьютеров в год.

Пятое поколение Характерной особенностью пятого поколения ЭВМ является то, что основные концепции этого поколения были заранее сформулированы в явном виде. Задача разработки принципиально новых компьютеров впервые поставлена в 1979 году японскими специалистами, объединившими свои усилия под эгидой научно-исследовательского центра по обработке информации - JPDEC. В 1981 г. JPDEC опубликовал предварительный отчет, содержащий детальный план развертывания научно-исследовательских и опытно-конструкторских работ с целью создания к 1991 г. прототипа ЭВМ нового поколения.

В начале 70-х годов произошел технологический прорыв в области производства компьютерных компонентов - появились большие интегральные схемы. Их сравнительно невысокая стоимость и высокие функциональные возможности привели к созданию мини-компьютеров, которые стали реальными конкурентами мэйнфреймов. Закон Гроша перестал соответствовать действительности, так как десяток мини-компьютеров выполнял некоторые задачи (как правило, хорошо распараллеливаемые) быстрее одного мэйнфрейма, а стоимость такой мини-компьютерной системы была меньше.

Но шло время, потребности пользователей вычислительной техники росли, им стало недостаточно собственных компьютеров, им уже хотелось получить возможность обмена данными с другими близко расположенными компьютерами. В ответ на эту потребность предприятия и организации стали соединять свои мини-компьютеры вместе и разрабатывать программное обеспечение, необходимое для их взаимодействия. В результате появились

первые локальные вычислительные сети . Они еще во многом отличались от современных локальных сетей, в первую очередь - своими устройствами сопряжения. На первых порах для соединения компьютеров друг с другом использовались самые разнообразные нестандартные устройства со своим способом представления данных на линиях связи, своими типами кабелей и т. п. Эти устройства могли соединять только те типы компьютеров, для которых были разработаны.

В середине 80-х годов утвердились стандартные технологии объединения компьютеров в сеть - Ethernet, Arcnet, Token Ring. Мощным стимулом для их развития послужили персональные компьютеры. Эти массовые продукты явились идеальными элементами для построения сетей - с одной стороны, они были достаточно мощными для работы сетевого программного обеспечения, а с другой - явно нуждались в объединении своей вычислительной мощности для решения сложных задач, а также разделения дорогих периферийных устройств и дисковых массивов. Поэтому персональные компьютеры стали преобладать в локальных сетях, причем, не только в качестве клиентских компьютеров, но и в качестве центров хранения и обработки данных, то есть сетевых серверов, потеснив с этих привычных ролей мини-компьютеры и мэйнфреймы.

Стандартные сетевые технологии превратили процесс построения локальной сети из искусства в рутинную работу. Для создания сети достаточно было приобрести сетевые адаптеры соответствующего стандарта, например Ethernet, стандартный кабель, присоединить адаптеры к кабелю стандартными разъемами и установить на компьютер одну из популярных сетевых операционных систем, например, NetWare. После этого сеть начинала работать и присоединение каждого нового компьютера не вызывало никаких проблем - естественно, если на нем был установлен сетевой адаптер той же технологии.

Последствием и одновременно движущей силой такого прогресса стало появление огромного числа непрофессиональных пользователей, которым совершенно не нужно было изучать специальные (и достаточно сложные) команды для сетевой работы. А возможность реализовать все эти удобства разработчики локальных сетей получили в результате появления качественных кабельных линий связи, на которых даже сетевые адаптеры первого поколения обеспечивали скорость передачи данных до 10 Мбит/с.

Контрольные вопросы:

1. Каковы цели и задачи изучения дисциплины?
2. Что изучает дисциплина?
3. Какие базовые знания нужны для изучения дисциплины?
4. Почему необходимо изучать курс «Архитектура компьютерных систем» студентам специальности «Информационные системы»?
5. Каковы современные тенденции развития средств вычислительной техники?
6. Этапы развития ВК, ВС и сетей.

Тема 1. Основы организации ЭВМ

1. Состав и назначение элементов компьютерных систем.
2. Классификация ЭВМ.
3. Основные устройства ЭВМ и их назначение. Архитектура, структура и интерфейсы ЭВМ.
4. Принцип программного управления работой ЭВМ.
5. Основные характеристики и параметры ЭВМ.

1 Состав и назначение элементов компьютерных систем.

Электронные вычислительные машины (ЭВМ) являются преобразователями информации. В них исходные данные задачи преобразуются в результат ее решения. В соответствии с используемой формой представления информации машин делятся на два больших класса - *аналоговые* и дискретного действия - *цифровые*. Их обозначают как АВМ и ЦВМ. В настоящее время термин ЭВМ относят именно к машине дискретного действия или ЦВМ, принципы, функционирования которых и будут рассмотрены в настоящем курсе.

Компьютер (ЭВМ) - это техническое устройство, предназначенное для обработки больших объёмов информации.

Под техническим устройством подразумевается объект искусственного происхождения, изготовленный специально с вполне определённым функциональным назначением.

Информацией называется совокупность данных произвольной природы, которые несут для пользователя вполне определённые сведения.

Данными называются материальные объекты, обладающие свойством отображать и нести на себе информацию. Их основное назначение заключается в том, чтобы материализовать информацию.

Обработка информации – её целенаправленное преобразование, то есть такое воздействие на информацию, которое оставляет различимые следы её преобразования. Автоматическая обработка информации характеризуется тем, что она выполняется в полном объёме без какого-либо участия человека.

Основные характеристики ЭВМ определяются характеристиками его компонентов. Каждый компонент представляет собой отдельное устройство, которое может как само по себе, так может и во взаимодействии с другими устройствами и определяет характеристики ЭВМ.

К основным компонентам ЭВМ относятся: центральный процессор, системная плата, основная память, монитор, жесткий диск, графическая карта, дисковод для компакт-дисков.

2 Классификация ЭВМ

Любая сфера человеческой деятельности, любой процесс функционирования технического объекта связаны с передачей и

преобразованием информации. Одно из важнейших положений кибернетики состоит в том, что без информации, ее передачи и переработки невозможны организованные системы – ни биологические, ни технические, искусственно созданные человеком.

В общем случае сообщения могут быть непрерывными (аналоговыми) и дискретными (цифровыми).

Аналоговое сообщение представляется некоторой физической величиной (обычно электрическим током или напряжением), изменение которой во времени отражает протекание рассматриваемого процесса, например температуры в нагревательной печи. Физический процесс, передающий непрерывное сообщение, может в определенном интервале принимать любые значения и изменяться в произвольные моменты времени, т.е. может иметь бесконечное множество состояний.

Дискретное сообщение характеризуется конечным набором состояний, например, передача текста. Каждое из этих состояний можно представить в виде конечной последовательности *символов* или *букв*, принадлежащих конечному множеству, называемому *алфавитом*. Такая операция называется *кодированием*, а последовательность символов – *кодом*. Число символов, входящих в алфавит, называется *основанием кода*. Важным здесь является не физическая природа символов кода, а то, что за конечное время можно передать только конечное число состояний сообщения. Причем, чем меньше основание кода, тем длиннее требуются кодовые группы для передачи фиксированного набора состояний сообщения.

В настоящее время в абсолютном большинстве случаев используются коды с основанием два, т.е. информация представляется в виде бинарных импульсных последовательностей, или *двоичных кодов*.

Передачу и преобразование любой дискретной информации всегда можно свести к эквивалентной передаче и преобразованию двоичных кодов, или *цифровой информации*.

Более того, возможно с любой заранее заданной степенью точности непрерывное сообщение заменить цифровым путем квантования непрерывного сообщения по уровню и дискретизации его по времени. Однако следует иметь в виду, что с увеличением точности представления аналогового сообщения растет разрядность кода. Это может привести к тому, что обработка аналогового сообщения в цифровой форме на конкретной ЭВМ в реальном масштабе времени окажется невозможной.

Таким образом, *любое* сообщение может быть с определенной степенью точности представлено в *цифровой* форме.

Электронные вычислительные машины (ЭВМ) являются преобразователями информации. В них исходные данные задачи преобразуются в результат ее решения. В соответствии с используемой формой представления информации при преобразовании ЭВМ делятся на два больших класса – *аналоговые* и дискретного действия – *цифровые*. Их обозначают как АВМ и ЦВМ соответственно. С 70-х годов термин ЭВМ относят именно к машинам

дискретного действия, или ЦВМ, принципы функционирования которых и будут рассмотрены в настоящем курсе.

3 Основные устройства ЭВМ и их назначение. Архитектура, структура и интерфейсы ЭВМ

Как уже отмечалось, современная ЭВМ есть вычислительный автомат, который представляет собой техническое устройство, способное автоматически исполнять заданные вычислительные алгоритмы. Данное назначение ЭВМ определяет её функциональную структуру и принцип действия.

Структурой какого-либо объекта принято называть его поэлементный состав с учётом существующих взаимосвязей между элементами. Рассмотрим функциональную структуру ЭВМ на уровне типовых укрупнённых функциональных блоков, то есть таких элементов конструкций, которые представляют собой основные самостоятельные устройства в составе ЭВМ (рисунок 1).

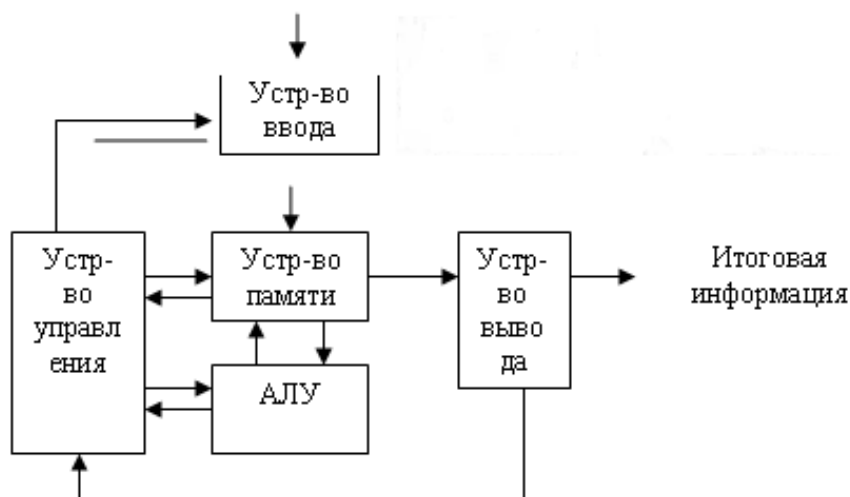


Рисунок 1 – Структура ЭВМ

Центральным устройством ЭВМ является устройство памяти (УП). Оно выполняет функцию накопления и хранения данных в процессе их обработки. Различают 2 вида памяти:

- А) Основная память;
- Б) Вспомогательная память.

Основная память включает следующие компоненты:

- 1) Оперативное запоминающее устройство (ОЗУ)
- 2) Постоянное запоминающее устройство (ПЗУ)
- 3) Программируемое (переменное) запоминающее устройство (ППЗУ)
- 4) Регистры
- 5) Аккумуляторы

Главным элементом основной памяти является ОЗУ. Оно выполняет функцию накопления и хранения данных оперативного доступа. Доступом называется произвольная операция обращения к памяти. Данные оперативного

доступа характеризуются тем, что могут изменять свои значения по ходу вычислительного процесса произвольное число раз. При этом требуют максимально возможной скорости обработки.

ОЗУ представляет собой основное рабочее место ЭВМ. При этом оно характеризуется тем, что имеет постоянный ограниченный объём, и этот объём в первую очередь определяет вычислительную производительность ЭВМ. ОЗУ обеспечивает самый короткий доступ к хранимым данным. Масштаб времени доступа составляет доли микросекунд.

ПЗУ представляет собой такой элемент памяти, в котором хранимая информация организована и представлена как элемент конструкции. Т. о. Данная информация является постоянной характеристикой устройства и не допускает своего изменения без вмешательства в конструкции, т.е. без разрушения самого устройства. ПЗУ выполняется в виде самостоятельной съёмной детали в структуре ЭВМ.

ППЗУ - такое ПЗУ, в котором существует возможность внесения изменений в хранимую информацию (перепрограммирование). Такое перепрограммирование выполняется с помощью специализированных устройств, называемых программаторами. Устройство типа ППЗУ ещё называют элементами флэш-памяти.

Регистром называется элемент рассредоточенной памяти ёмкостью, соизмеримой с размером одного цифрового машинного слова. Обычно регистры выполняют функцию «персонального обслуживания» устройства, за которым он закреплён. В современных моделях ЭВМ регистры образуют внутреннюю индивидуальную память микропроцессора.

Аккумулятором называется специализированный регистр, предназначенный для прямого обслуживания АЛУ.

Вспомогательная (внешняя) память выполняет функцию накопления и хранения информации архивного назначения. Архивная информация характеризуется тем, что:

1) Предназначается для длительного хранения, т.е. для хранения такой промежутков времени, который превосходит длительность одного отдельно взятого вычислительного процесса.

2) Данные в составе архивной информации ориентированы на использование не в одной отдельно взятой задаче, а в различных задачах.

3) Архивные данные допускают обращение к ним произвольное количество раз, причём из различных задач. При этом обращение к данным не допускает никакого изменения этих данных. Т.е. к таким данным разрешено обращение только на чтение.

4) Архивная информация предполагает достаточно большой объём, т.е. такой объём, который покрывает все возникающие потребности. Отсюда следует, что размер внешней памяти для хранения архива должен быть практически ничем не ограничен.

Внешняя память конструктивно выполняется в виде накопителей информации. Такие накопители характеризуются тем, что обеспечивают

возможность закрепления (записи) информации на специальных машинных носителях.

Для измерения ёмкости памяти ЭВМ используются специальные единицы, основной из которой является 1 байт. 1 байт – это такое место в памяти ЭВМ, куда может быть размещён 1 элементарный символ информации. Такой символ не может быть разделён на составляющие без потери признаков символа.

Для современных ЭВМ со средними показателями вычислительной производительности характерны следующие показатели ёмкости памяти:

Ёмкость ОЗУ -128-256 МВ

Ёмкость винчестера – 1ГВ

Ёмкость стандартного гибкого диска с определяющим размером 3,5 дюйма – 1,44 МВ.

Ёмкость регистра памяти соизмеряется с длиной 1 цифрового машинного слова, т.е. цифровой комбинации, заданной стандартной длиной в байтах.

Арифметико-логическое устройство. АЛУ выполняет функцию воспроизведения заданной элементарной вычислительной операции над заданными данными.

Вычислительной операцией называется некое вычислительное действие над данными. Всякая операция, предназначенная для исполнения на ЭВМ, должна быть предварительно описана в форме записи. Запись операции характеризуется тем, что представляет собой совокупность некоторого числа данных, соединённых символами вычислительных операций. Операции по масштабу делятся на составные и элементарные. Составные операции характеризуются тем, что они всегда могут быть разбиты на отдельные элементарные операции, тогда как элементарные операции не могут быть разбиты на составляющие без потери признаков операции.

Устройство управления (УУ). УУ выполняет функцию обеспечения совместной согласованной работы всех устройств ЭВМ.

В современных ЭВМ АЛУ и УУ конструктивно объединены в одном функциональном блоке, который называется процессором. Такой процессор, выполненный по микроэлектронной технологии, называется микропроцессором. Т.о процессор в ЭВМ представляет собой собственно автоматическое вычислительное устройство. Его технические характеристики определяют вычислительную производительность ЭВМ. Основными такими характеристиками являются показатели разрядности и тактовой частоты.

Разрядностью процессора называется количество информации, которое способен обработать процессор одновременно за 1 акт (т.е. за 1 движение). Тактовая частота - это количество тактов, которое вырабатывает устройство отсчёта времени ЭВМ (следовательно, микропроцессора) в 1 физическую секунду. 1 такт - это минимальная порция времени, различаемая устройствами ЭВМ. Тактовая частота измеряется в Гц. 1 Гц =1 такт/сек. На практике пользуются производной единицей 1 МГц-10⁶ такт/сек.

Т.о., показатель разрядности определяет порцию одновременно обрабатываемой информации, а тактовая частота определяет скоростные

характеристики микропроцессора и ЭВМ в целом. Показатель разрядности оценивается числом двоичных разрядов (числом бит).

Устройство ввода выполняет функцию загрузки памяти ЭВМ (ОЗУ) исходной информации для счёта. Исходной информацией считается такая, которая должна быть задана и определена на момент начала вычислительного процесса, т.к. без этого вычислительный процесс не может начаться. Функция ввода организуется на основе автоматического преобразования информации домашинной формы представления в информацию внутримашинного представления с последующим размещением преобразованной информации в ОЗУ. В современных ЭВМ в качестве стандартных устройств ввода используются клавиатура и накопители внешней памяти в режиме чтения. Стандартные устройства- это такие устройства, которые входят в обязательный комплект поставки ЭВМ.

Устройство вывода (УВ) выполняет функцию выгрузки из памяти ЭВМ, отображения их в соответствии с требованиями и при необходимости закрепления (записи) на технический носитель для цели длительного хранения. Эта функция выполняется на основе преобразования информации внутримашинной формы представления во внешнюю форму.

Магистрально-модульный принцип организации ЭВМ получил широкое распространение, поскольку в этом случае все устройства используют единый подход сопряжения модулей центральных процессоров и устройств ввода/вывода с помощью стандартных шин.

Традиционно шины делятся на шины, обеспечивающие организацию связи процессора с памятью, и шины ввода/вывода. Шины ввода/вывода могут иметь большую протяженность, поддерживать подсоединение многих типов устройств и обычно следуют одному из шинных стандартов. Шины процессор-память, с другой стороны, сравнительно короткие, обычно высокоскоростные и соответствуют организации системы памяти для обеспечения максимальной пропускной способности канала память-процессор. На этапе разработки системы для шины процессор-память заранее известны все типы и параметры устройств, которые должны соединяться между собой, в то время как разработчик шины ввода/вывода должен иметь дело с устройствами, различающимися по задержке и пропускной способности.

С целью снижения стоимости некоторые компьютеры снабжаются единственной шиной для памяти и устройств ввода/вывода. Такая шина часто называется системной. Персональные компьютеры, как правило, строятся на основе одной системной шины.

Необходимость сохранения баланса производительности по мере роста быстродействия микропроцессоров привела к двухуровневой организации шин в персональных компьютерах на основе локальной шины.

Локальной шиной называется шина, выходящая непосредственно на контакты микропроцессора. Она обычно объединяет процессор, память, схемы буферизации для системной шины и ее контроллер, а также некоторые вспомогательные схемы. Типичными примерами локальных шин являются VL-Bus и PCI.

На современных компьютерах часто используется специальная магистраль обмена данными с процессором – локальная видеошина. AGP (Accelerated Graphics Port - ускоренный графический порт) – скоростная шина для связи с графической картой. Разработана Intel для высокоскоростной графики. Основное преимущество этой шины заключается в скорости.

Различают шины синхронные и асинхронные. Синхронная шина использует сигналы синхронизации, которые передаются по линиям управления шины, и фиксированный протокол, определяющий расположение сигналов адреса и данных относительно сигналов синхронизации. Эти шины являются и быстрыми, и дешевыми. Однако они имеют два главных недостатка. Все на шине должно происходить с одной и той же частотой синхронизации, поэтому из-за проблемы перекося синхросигналов синхронные шины не могут быть длинными. Обычно такие шины связывают процессор и память.

Асинхронная шина не тактируется. Вместо этого обычно используется старт-стопный режим передачи и протокол «рукопожатия» (handshaking) между источником и приемником данных на шине. Эта схема позволяет гораздо проще приспособить широкое разнообразие устройств и удлинить шину без беспокойства о перекося сигналов синхронизации и о системе синхронизации. Асинхронные шины по мере изменения технологии лучше масштабируются. Они обычно используются в устройствах ввода/вывода.

Стандарты шин. Обычно количество и типы устройств ввода/вывода в вычислительных системах не фиксируются, что позволяет пользователю самому подобрать необходимую конфигурацию. Шина ввода/вывода компьютера может рассматриваться как шина расширения, обеспечивающая постепенное наращивание устройств ввода/вывода. Поэтому стандарты играют огромную роль, позволяя разработчикам компьютеров и устройств ввода/вывода работать независимо. Появление стандартов определяется разными обстоятельствами.

Иногда широкое распространение и популярность конкретных машин становятся причиной того, что их шина ввода/вывода становится стандартом де-факто. Примерами таких шин могут служить PDP-11 Unibus и IBM PC/AT Bus. Иногда стандарты появляются также в результате определенных достижений по стандартизации в некотором секторе рынка устройств ввода/вывода. Интеллектуальный периферийный интерфейс (IPI - Intelligent Peripheral Interface) и Ethernet являются примерами стандартов, появившихся в результате кооперации производителей. Успех того или иного стандарта в значительной степени определяется его принятием такими организациями как ANSI (Национальный институт по стандартизации США) или IEEE (Институт инженеров по электротехнике и радиоэлектронике). Иногда стандарт шины может быть прямо разработан одним из комитетов по стандартизации: примером такого стандарта шины является FutureBus.

Одной из популярных шин персональных компьютеров была системная шина *IBM PC/XT*, обеспечивавшая передачу 8 бит данных. Кроме того, эта шина включала 20 адресных линий, которые ограничивали адресное пространство пределом в 1 Мбайт. Для работы с внешними устройствами в этой шине были предусмотрены также 4 линии аппаратных прерываний (IRQ) и

4 линии для требования внешними устройствами прямого доступа к памяти (DMA). Для подключения плат расширения использовались специальные 62-контактные разъемы. При этом системная шина и микропроцессор синхронизировались от одного тактового генератора с частотой 4.77 МГц. Таким образом, теоретическая скорость передачи данных могла достигать немногим более 4 Мбайт/с.

Системная шина *ISA* (Industry Standard Architecture) впервые стала применяться в персональных компьютерах IBM PC/AT на базе процессора 80286. Для этой системной шины характерно наличие второго, 36-контактного дополнительного разъема для соответствующих плат расширения. За счет этого количество адресных линий было увеличено на 4, а данных - на 8, что позволило передавать параллельно 16 бит данных и обращаться к 16 Мбайт системной памяти. Количество линий аппаратных прерываний в этой шине было увеличено до 15, а каналов прямого доступа - до 7. Системная шина *ISA* полностью включала в себя возможности старой 8-разрядной шины. Шина *ISA* позволяет синхронизировать работу процессора и шины с разными тактовыми частотами. Она работает на частоте 8 МГц, что соответствует максимальной скорости передачи 16 Мбайт/с.

С появлением процессоров 80386, 80486 и Pentium шина *ISA* стала узким местом персональных компьютеров на их основе. Новая системная шина *EISA* (Extended Industry Standard Architecture), появившаяся в конце 1988 года, обеспечивает адресное пространство в 4 Гбайта, 32-битовую передачу данных (в том числе и в режиме DMA), улучшенную систему прерываний и арбитраж DMA, автоматическую конфигурацию системы и плат расширения. Устройства шины *ISA* могут работать на шине *EISA*. Шина *EISA* предусматривает централизованное управление доступом к шине за счет наличия специального устройства - арбитра шины. Поэтому к ней может подключаться несколько главных устройств шины. Улучшенная система прерываний позволяет подключать к каждой физической линии запроса на прерывание несколько устройств, что снимает проблему количества линий прерывания. Шина *EISA* тактируется частотой около 8 МГц и имеет максимальную теоретическую скорость передачи данных 33 Мбайт/с.

Шина *MCA* также обеспечивает 32-разрядную передачу данных, тактируется частотой 10 МГц, имеет средства автоматического конфигурирования и арбитража запросов. В отличие от *EISA* она не совместима с шиной *ISA* и используется только в компьютерах компании IBM.

Шина *VL-bus*, предложенная ассоциацией VESA (Video Electronics Standard Association), предназначалась для увеличения быстродействия видеоадаптеров и контроллеров дисковых накопителей для того, чтобы они могли работать с тактовой частотой до 40 МГц. Шина *VL-bus* имеет 32 линии данных и позволяет подключать до трех периферийных устройств, в качестве которых наряду с видеоадаптерами и дисковыми контроллерами могут выступать и сетевые адаптеры. Максимальная скорость передачи данных по шине *VL-bus* может составлять около 130 Мбайт/с.

Шина PCI (Peripheral Component Interconnect) также, как и шина VL-bus, поддерживает 32-битовый канал передачи данных между процессором и периферийными устройствами, работает на тактовой частоте 33 МГц и имеет максимальную пропускную способность 120 Мбайт/с. При работе с процессорами 80486 шина PCI дает примерно те же показатели производительности, что и шина VL-bus. Однако, в отличие от последней, шина PCI является процессорно независимой (шина VL-bus подключается непосредственно к процессору 80486 и только к нему). Ее легко подключить к различным центральным процессорам. В их числе Pentium, Alpha, R4400 и PowerPC.

Основные возможности шины PCI:

- синхронный протокол обмена, предусматривающий подтверждение о приеме, что позволяет пересылать данные со скоростью, максимально возможной для приемного устройства (в отличие от ISA);
- в PC используется 32-разрядная шина PCI с частотой 33 МГц. Однако шина PCI процессорно независимая, и спецификация допускает ширину в 64 разряда и частоту 66 МГц. Это используется, например, в компьютерах компании Sun, в серверах на базе ЦП Alpha и Intel Xeon. Слот 32-разрядной шины PCI имеет 124 контакта (у 64-разрядной есть еще дополнительная секция);
- вставляемые платы автоматически конфигурируются (при старте) по стандарту Plug and Play, т.е. не надо вручную устанавливать перемычки и переключатели, чтобы избежать конфликтов с другими устройствами;
- для пользователя шина предоставляет слоты (не более пяти). Это так называемый сегмент PCI;
- спецификация PCI допускает произвольное число сегментов, образующих древовидную топологию. Для соединения сегментов используются PCI-мосты;
- есть режим Bits mastering, дающий устройству возможность брать управление шиной на себя и тем самым разгрузить ЦП;
- начиная с версии 2.2 (внедрена в 1999 г.) в шину включена линия, управляющая пробуждением компьютера от PCI-устройства, например, модема, сетевой карты.

PCI Special Interest Group в октябре 1999 г. объявила о выходе спецификации PCI-X 1.0. Спецификация представляет собой окончательный вариант преемницы PCI-шины. Основным достоинством новой шины является увеличенная до 133 МГц частота при сохранившейся обратной совместимости с PCI-картами.

Одной из популярных шин ввода-вывода является шина SCSI. Под термином SCSI - Small Computer System Interface (Интерфейс малых вычислительных систем) обычно понимается набор стандартов, разработанных Национальным институтом стандартов США (ANSI) и определяющих механизм реализации магистрали передачи данных между системной шиной компьютера и периферийными устройствами.

Первоначально SCSI предназначался для использования в небольших дешевых системах и поэтому был ориентирован на достижение хороших результатов при низкой стоимости. Характерной его чертой является простота, особенно в части обеспечения гибкости конфигурирования периферийных устройств без изменения организации основного процессора. Главной особенностью подсистемы SCSI является размещение в периферийном оборудовании интеллектуального контроллера.

Для достижения требуемого высокого уровня независимости от типов периферийных устройств в операционной системе основной машины, устройства SCSI представляются имеющими очень простую архитектуру. Например, геометрия дискового накопителя представляется в виде линейной последовательности одинаковых блоков, хотя в действительности любой диск имеет более сложную многомерную геометрию, содержащую поверхности, цилиндры, дорожки, характеристики плотности, таблицу дефектных блоков и множество других деталей. В этом случае само устройство или его контроллер несут ответственность за преобразование упрощенной SCSI модели в данные для реального устройства.

Начальный стандарт 1986 года, известный теперь под названием SCSI-1, определял рабочие спецификации протокола шины, набор команд и электрические параметры. Шина данных SCSI-1 использовала 50-жильный экранированный кабель, имела разрядность 8 бит, а максимальная скорость передачи составляла 5 Мбайт/сек.

В 1992 году этот стандарт был пересмотрен с целью устранения недостатков первоначальной спецификации (особенно в части синхронного режима передачи данных) и добавления новых возможностей повышения производительности, таких как «быстрый режим» (fast mode), «широкий режим» (wide mode) и помеченные очереди. Этот пересмотренный стандарт получил название SCSI-2 и в настоящее время используется большинством поставщиков вычислительных систем.

Стандарт SCSI-2 определяет, в частности, различные режимы: Wide SCSI, Fast SCSI и Fast-and-Wide SCSI. Fast SCSI сохраняет 8-битовую шину данных и поэтому может использовать те же самые физические кабели, что и SCSI-1. Он отличается только тем, что допускает передачи со скоростью 10 Мбайт/сек в синхронном режиме. Wide SCSI удваивает либо учетверяет разрядность шины данных (либо 16, либо 32 бит), допуская соответственно передачи со скоростью либо 10, либо 20 Мбайт/сек. В комбинации Fast-and-Wide SCSI возможно достижение скоростей передачи 20 и 40 Мбайт/сек соответственно.

Однако поскольку в обычном 50-жильном кабеле просто не хватает жил, комитет SCSI решил расширить спецификацию вторым 66-жильным кабелем (так называемый В-кабель). В-кабель имеет дополнительные линии данных и ряд других сигнальных линий, позволяющие реализовать режим Fast-and-Wide.

В реализации режима Wide SCSI предложена также расширенная адресация, допускающая подсоединение к шине до 16 устройств (вместо стандартных восьми). Это значительно увеличивает гибкость подсистемы SCSI,

правда, приводит к появлению дополнительных проблем, связанных с эффективностью ее использования.

Реализация режимов Wide-SCSI и Fast-and-Wide SCSI до 1994 года редко использовалась, поскольку эффективность их применения не была достаточно высокой. Однако широкое распространение дисковых массивов и дисковых накопителей со скоростью вращения 7200 оборотов в минуту делают эту технологию весьма актуальной.

4 Принцип программного управления работой ЭВМ

Для того, чтобы понять принцип программного управления ЭВМ, необходимо определить понятия *алгоритм* и *программа*. Понятие алгоритма не замыкается только областью вычислительной техники (ВТ). По интуитивному определению:

Алгоритм – это совокупность правил, строго следуя которым можно перейти от исходных данных к конечному результату.

В вычислительной технике под "совокупностью правил" понимается последовательность арифметических и логических операций. (Утверждают, что слово *алгоритм* произошло от имени Мухаммед аль Хорезми, написавшем в IX веке трактат по арифметике десятичных чисел.)

Программа – это запись алгоритма в форме, воспринимаемой ЭВМ. Любая программа состоит из отдельных *команд*. Каждая команда предписывает определенное действие и указывает, над какими операндами это действие производится. Программа представляет собой *совокупность* команд, записанных в определенной последовательности, обеспечивающих решение задачи на ЭВМ. Для того, чтобы УУ могло воспринять команды, они должны быть закодированы в цифровой форме (во всех современных ЭВМ – это двоичный код).

Автоматическое управление процессом решения задачи достигается на основе *принципа программного управления*, являющегося основной особенностью ЭВМ. (Без программного управления ЭВМ превратится в обычный быстродействующий арифмометр или калькулятор.)

Другим важнейшим принципом является *принцип хранимой в памяти программы*. Согласно этому принципу программа, закодированная в цифровом виде, хранится в памяти наравне с числами. Поскольку программа хранится в памяти, одни и те же команды можно извлекать и выполнять необходимое количество раз. Более того, над кодами команд можно выполнять некоторые арифметические операции и тем самым *модифицировать* адреса обращения к ОП.

Команды программы выполняются в порядке, соответствующем их расположению в последовательных ячейках памяти. Однако команды *безусловного* и *условного переходов* могут изменять этот порядок, соответственно безусловно, или при выполнении некоторого условия, задаваемого отношениями типа больше, меньше или равно. В большинстве случаев сравниваются результаты выполнения предыдущей операции и

некоторое число, указанное в команде условного перехода. Именно команды условного перехода позволяют строить не только линейные, но также ветвящиеся и циклические программы.

5 Основные характеристики и параметры ЭВМ

Структуру ЭВМ определяет следующая группа характеристик:

- технические и эксплуатационные характеристики ЭВМ (быстродействие и производительность, показатели надежности, достоверности, точности, емкость оперативной и внешней памяти, габаритные размеры, стоимость технических и программных средств, особенности эксплуатации т.д.);
- характеристики и состав функциональных модулей базовой конфигурации ЭВМ; возможность расширения состава технических и программных средств; возможность изменения структуры;
- состав программного обеспечения ЭВМ и сервисных услуг (операционная система или среда, пакеты прикладных программ, средства автоматизации программирования).

К основным характеристикам ЭВМ относятся:

Быстродействие это число команд, выполняемых ЭВМ за одну секунду.

Сравнение по быстродействию различных типов ЭВМ, не обеспечивает достоверных оценок. Очень часто вместо характеристики быстродействия используют связанную с ней характеристику производительность.

Производительность это объем работ, осуществляемых ЭВМ в единицу времени.

Применяются также относительные характеристики производительности. Фирма Intel для оценки процессоров предложила тест, получивший название индекс iCOMP (Intel Comparative Microprocessor Performance). При его определении учитываются четыре главных аспекта производительности: работа с целыми числами, с плавающей запятой, графикой и видео. Данные имеют 16- и 32-разрядное представление. Каждый из восьми параметров при вычислении участвует со своим весовым коэффициентом, определяемым по усредненному соотношению между этими операциями в реальных задачах. По индексу iCOMP ПМ Pentium 100 имеет значение 810, а Pentium 133-1000.

Емкость запоминающих устройств. Емкость памяти измеряется количеством структурных единиц информации, которое может одновременно находиться в памяти. Этот показатель позволяет определить, какой набор программ и данных может быть одновременно размещен в памяти.

Наименьшей структурной единицей информации является **бит**- одна двоичная цифра. Как правило, емкость памяти оценивается в более крупных единицах измерения - байтах (байт равен восьми битам). Следующими единицами измерения служат 1 Кбайт = 2¹⁰ = 1024 байта, 1 Мбайт = 2¹⁰ Кбайт = 2²⁰ байта, 1 Гбайт = 2¹⁰ Мбайт = 2³⁰ байта.

Емкость оперативной памяти (ОЗУ) и емкость внешней памяти (ВЗУ) характеризуются отдельно. Этот показатель очень важен для определения,

какие программные пакеты и их приложения могут одновременно обрабатываться в машине.

Надежность это способность ЭВМ при определенных условиях выполнять требуемые функции в течение заданного периода времени (стандарт ISO (Международная организация стандартов) 2382/14-78).

Высокая надежность ЭВМ закладывается в процессе ее производства. Применение сверхбольшие интегральные схемы (СБИС) резко сокращают число используемых интегральных схем, а значит, и число их соединений друг с другом. Модульный принцип построения позволяет легко проверять и контролировать работу всех устройств, проводить диагностику и устранение неисправностей.

Точность это возможность различать почти равные значения (стандарт ISO - 2382/2-76).

Точность получения результатов обработки в основном определяется разрядностью ЭВМ, а также используемыми структурными единицами представления информации (байтом, словом, двойным словом).

Достоверность это свойство информации быть правильно воспринятой.

Достоверность характеризуется вероятностью получения безошибочных результатов. Заданный уровень достоверности обеспечивается аппаратурно-программными средствами контроля самой ЭВМ. Возможны методы контроля достоверности путем решения эталонных задач и повторных расчетов. В особо ответственных случаях проводятся контрольные решения на других ЭВМ и сравнение результатов.

Контрольные вопросы:

1. Какие элементы входят в состав компьютерных систем?
2. Объясните их назначение.
3. Как классифицируются ЭВМ в соответствии с используемой формой представления информации при ее преобразовании?
4. Поясните структуру ЭВМ,
5. Какие устройства входят в архитектуру ЭВМ?
6. В чем выражается сущность магистрально-модульного принципа построения ЭВМ?
7. Материнские платы.
8. Структура чипсетов.
9. Перечислите основные характеристики ЭВМ.

Тема 2. Представление информации в ЭВМ. Арифметические и логические основы ЭВМ

1. Представление информации в ЭВМ. Арифметические и логические основы ЭВМ.
2. Основы теории логического проектирования цифровых устройств.
3. Базовые элементы ЭВМ, состав и характеристики.

1 Представление информации в ЭВМ. Арифметические и логические основы ЭВМ

Внутримашинное представление информации соответствует её отображению в виде цифровых кодов. В результате все действия ЭВМ характеризуются как вычислительные, т.е. такие, которые выполняются в форме прямых операций над числами. Отсюда следует, что одним из основополагающих понятий вычислительной техники является понятие числа.

Числом называется количественная характеристика объекта, отвечающая на вопрос «сколько».

При графическом способе отображения чисел вводится определённая символика в виде системы символов и правил использования этих символов для отображения значений числа. Такая символика позволяет только отображать числа, но не обеспечивает возможностей отображения операций над числами и возможностей определения результатов этих операций. Все эти возможности обеспечиваются системой счисления.

До настоящего времени известны 2 основных способа организации систем счисления. В соответствии с первым способом организуются так называемые символьные системы счисления, а в соответствии со 2-ым способом позиционные системы счисления. И в тех, и в других системах базовым понятием является цифра.

Цифрой называется элементарный символ, используемый для отображения числа. Всякое число отображается как составная конструкция в виде последовательности цифр. В символьной системе счисления каждая цифра имеет одно и то же значение вне зависимости от того, какое место оно занимает в общей записи числа. Примером символьной системы счисления является известная римская система.

В современной практике повсеместно используются позиционные системы счисления. Они характеризуются тем, что цифра в составе записи числа имеет различное значение в зависимости от того, какое место (позицию) оно занимает в общей записи числа.

Например: $333111 = 3 \cdot 10^5 + 3 \cdot 10^4 + 3 \cdot 10^3 + 1 \cdot 10^2 + 1 \cdot 10^1 + 1 \cdot 10^0$

Различные способы отображения числа предполагают понятия «отображённое число» и «истинное значение числа».

Отображённое число – это есть число, представленное с использованием определённой символики в определённой системе счисления.

Истинное значение числа есть само число как количественная характеристика объекта.

Среди всех позиционных систем истинное значение числа, совпадающее с его отображением, обеспечивает так называемая десятичная система счисления. Поэтому данная система нашла самое широкое применение и повсеместно используется в обычной практике выполнения вычислительных операций. Во всех других системах счисления запись числа не совпадает с истинным значением числа, в связи с чем, возникает необходимость вычисления истинного значения числа. Кроме того, возникает потребность в выполнении межсистемных преобразованиях чисел, т.е. перевода чисел из одних систем счисления в другие и выполнения эквивалентных операций над числами в различных системах счисления.

Позиционные системы счисления. В позиционной системе число A записывается в виде последовательности цифр, то есть имеет вид:

$$A = a_1 a_2 \dots a_n \cdot a_{n+1} a_{n+2} \dots a_r$$

Здесь группа цифр a_1 - a_n составляет целую часть числа, группа цифр a_{n+1} - a_r составляют дробную часть числа. Границу записи целой и дробной части определяет десятичная точка (символ). В данной записи каждая цифра в составе числа имеет значение, которое определяется позицией цифры в записи числа. Указанные позиции описываются по порядку влево и вправо от последней цифры целой части числа, которое имеет нулевую позицию. При этом нумерация позиции влево осуществляется с положительным знаком, а вправо с отрицательным знаком. С учётом произведённой нумерации позиции истинное значение числа A определяется как

$$|A| = a_1 p^{n-1} a_2 p^{n-2} \dots a_n p^0, a_{n+1} p^{-1} a_{n+2} p^{-2} \dots a_r p^{-r}$$

где p – основание системы исчисления.

Основанием системы исчисления называется число, которое определяет количество цифр, используемых в данной системе исчисления для отображения произвольных чисел. Указанные цифры образуют базовый цифровой ряд позиционной системы исчисления. В десятичной системе исчисления базовый цифровой ряд образуют цифры 0,1,2,3,4,5,6,7,8,9. При этом основанием системы исчисления является число 10.

В позиционной системе исчисления произвольное число может быть представлено двумя возможными способами:

- а) способ с фиксированной точкой;
- б) способ с плавающей точкой.

Числом a с фиксированной точкой называется такое число a со знаком, которое содержит целую и дробную часть, разделённые символом «десятичная точка». Способ представления числа с фиксированной точкой характеризуется тем, что значение числа размещается на цифровой разрядной сетке установленного формата. Данный формат определяет общую длину разрядной сетки, длину целой и дробной частей, а также положение десятичной точки,

которое фиксируется в отдельном разряде. При этом запись значения числа осуществляется соответственно влево и вправо от десятичной точки. При таком способе представления числа возникают понятия «минимально представимые» значения в заданном формате и максимально представимое значение в заданном формате.

Числом, минимально представимым на заданном формате, называется такое число, которое содержит единицу в младшем дробном разряде, тогда как все остальные разряды заняты нулём.

Максимально представимое число в заданном формате характеризуется тем, что все цифровые разряды разрядной сетки заполнены старшей цифрой базового цифрового ряда используемой системы исчисления.

Всякое число, значение которого меньше минимально представимого, автоматически приравнивается к 0, который в данном случае называется машинным. Всякое число, значение которого превосходит максимально представимое, считается несуществующим и вызывает переполнение разрядной сетки, что является основанием для аварийного прекращения счёта.

Числом A с плавающей точкой называется такое число, запись которого имеет вид:

$$A=MEP$$

где M – мантисса числа; P – порядок числа; E – признак числа с плавающей точкой.

Например: 1.3987E2

Мантиссой числа с плавающей точкой называется число со знаком в виде числа с фиксированной точкой, на котором определяется истинное значение числа путём определения положения десятичной точки.

Порядок числа P – это есть число с фиксированной точкой целого типа, которое определяет положение десятичной точки на мантиссе числа при отсчёте на этой мантиссе истинного значения числа.

Истинное значение числа с плавающей точкой определяется как

$$|A|=M*S^P$$

где S – основание системы исчисления.

Таким образом, при использовании формата с плавающей точкой на одной и той же мантиссе может быть воспроизведено бесконечное количество чисел, отличающихся на порядок. Эти числа воспроизводятся путём изменения значения p на единицу, что вызывает эффект точки, плавающей по мантиссе. Диапазон представлений чисел способом с плавающей точкой значительно превосходит диапазон представления чисел способом с фиксированной точкой при использовании одинаковой цифровой разрядной сетки для представления числа.

Представление числа способом с плавающей точкой предполагает обязательное использование мантиссы числа. При этом в общем случае не оговаривается способ организации такой мантиссы, в связи с чем одно и то же число может иметь бесконечное число вариантов записи. С тем, чтобы

исключить элементы неоднозначности представления чисел вводится понятие нормализованной мантиссы.

Нормализованная мантисса - это такая мантисса, которая приведена к виду заранее обусловленного формата. Различают 2 основных формата нормализованных мантисс:

А) Математически нормализованная мантисса.

Б) Машинно-нормализованная мантисса.

Математически нормализованная мантисса характеризуется тем, что имеет нулевую целую часть, при этом старший разряд дробной части всегда представлен значащей цифрой. Например, число 375.05, представленное способом с плавающей точкой на математически нормализованной мантиссе, будет иметь вид: 0.37505E3.

Машинно-нормализованная мантисса - имеет целую часть в виде одной значащей цифры. В данном случае число 375.05 будет иметь вид: 3.7505E2.

Примечание: Обычно нормализация мантиссы предполагает определённые нормы обеспечения длины записи мантиссы. Так, представление числовых результатов при компьютерном счёте обычно осуществляется способом с плавающей точкой на машинно-нормализованной мантиссе, длина которой определяется моделью ЭВМ и версией ОС.

Двоичная система счисления. Современные средства цифровой вычислительной техники можно отнести к классу технических устройств, в которых реализована двоичная система счисления. Это связано с тем, что представление любого числа в двоичной системе основывается на использовании всего 2-ух цифр: 0 и 1, которые могут быть воспроизведены устройствами простейшей конструкции типа двухпозиционного переключателя.

Двоичная позиционная система счисления характеризуется тем, что её базовый цифровой ряд включает 2 цифры: 0 и 1. При этом основание системы счисления $S=2$.

Таблица соответствия основных десятиричных чисел двоичным числам

Дес. система	Дв. система
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100

13	1101
14	1110
15	1111
16	10000 В

Арифметика двоичных чисел. Выполнение арифметических операций над двоичными числами выполняется в соответствии со следующими значениями результатов операций:

$$\begin{array}{cccc}
 0+0=0 & 0-0=0 & 0*0=0 & 0:0=\text{н/о} \\
 0+1=1 & 1-0=1 & 1*0=0 & 1:0=\text{н/о} \\
 1+0=1 & 0-1=\text{н/о} & 0*1=0 & 0:1=0 \\
 1+1=10 \text{ В} & 1-1=0 & 1*1=1 & 1:1=1 \\
 & 10-1=1 & &
 \end{array}$$

Двоичное дополнение до двух. Запись всякого числа в любой системе счисления по существу можно расценивать как кодирование числа. Существуют различные способы такого кодирования. В вычислительной технике, в особенности, при работе с двоичными числами широко используется представление числа в виде прямого кода, обратного кода и дополнительного кода.

Прямой код числа характеризуется тем, что число представляется способом с фиксированной точкой. В случае обратного кода – в записи числа цифра 0 заменяется на 1 и наоборот – 1 на 0. Дополнительный код числа образуется путем добавления единицы в младший разряд обратного кода числа.

С использованием дополнительного кода числа можно организовать операцию вычитания чисел в форме операции сложения. Для этого оба числа, участвующие в операции вычитания, преобразуются эквивалентным образом к дробному значению с нулевой целой частью и приводятся к одному формату по длине. Недостающие цифровые разряды в вычитаемом заполняются нулями. Затем для вычитаемого образуется дополнительный код, который складывается с уменьшаемым. Результат операции отсчитывается на тех же цифровых разрядах, что и числа, участвующие в операции. Значение целой части в результате операции отбрасывается. Полученное дробное число преобразуется в исходный формат.

Кодированные вычисления могут производиться с использованием вспомогательных систем счисления, в частности восьмеричной и шестнадцатеричной позиционных систем.

Восьмеричная система исчисления. Перевод чисел из десятичной системы в двоичную требует выполнения достаточно большого объема вычислений. Вместе с тем, программирование вычислительной техники на практике требует организации вычислений всех видов только в двоичной системе. Кроме того, адресация памяти ЭВМ также выполняется только в двоичных кодах. Работа с такими кодами, вследствие их большой длины, обычно затруднена для запоминания значений данных. В связи с этим широкое применение находит восьмеричная система исчисления, которая позволяет выполнять двоично-восьмеричные преобразования без промежуточных

вычислений и таким образом облегчает запоминание длинных двоичных кодов. При этом вычислительные операции над двоичными числами могут быть заменены операциями над эквивалентными восьмеричными числами.

Преобразование восьмеричных чисел в двоичные выполняется путём прямого кодирования каждой восьмеричной цифры в составе числа двоичным кодом длиной в три разряда.

Преобразование двоичного числа в восьмеричное выполняется путём разбиения двоичного числа на группы по 3 разряда влево и вправо от десятичной точки с последующим переводом каждого трёхразрядного кода в восьмеричную цифру.

Базовый цифровой ряд восьмеричной системы включает цифры: 0,1,2,3,4,5,6,7. Соответственно, основанием системы исчисления является число 8.

Таблица соответствия между десятичными и восьмеричными числами

Дес. сист.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Восм. Сист.	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20

Преобразование восьмеричного числа в десятичное выполняется путём вычисления истинного значения восьмеричного числа.

Арифметика восьмеричных чисел. Арифметические операции над восьмеричными числами выполняются также как и над десятичными числами с той лишь разницей, что результат операции оценивается в абсолютных значениях с последующим переводом этих значений в восьмеричный код.

Представление чисел в восьмеричной системе способом с фиксированной и плавающей точкой выполняется по общим правилам для позиционных чисел. Для того, чтобы получить число с фиксированной точкой, достаточно заменить десятичную запятую символом точка. В случае записи числа представленного способом с плавающей точкой, мантисса числа записывается в виде восьмеричного числа, а порядок в виде десятичного числа.

Шестнадцатеричная система исчисления. В практике программирования ЭВМ использование восьмеричной системы бывает не всегда удобным, В особенности в тех случаях, когда выполняется занесение числовых значений данных в память ЭВМ, поскольку каждый цифровой разряд восьмеричного числа кодируется на трех двоичных разрядах. При занесении такого числа в память ЭВМ его поразрядная длина всегда кратна трем. Поэтому при его размещении в ячейках памяти в виде байтов, длина которых равна восьми двоичным разрядам, происходит потеря незанятых двоичных разрядов, что приводит к нерациональному использованию памяти ЭВМ.

В связи с этим обычно используется шестнадцатеричная система, которая обладает всеми свойствами восьмеричной системы с той разницей, что каждая шестнадцатеричная цифра кодируется на четырех двоичных разрядах. Поэтому при размещении такого числа в памяти ЭВМ в тех случаях, когда его поразрядная длина содержит четное количество цифровых разрядов, оно

занимает целое число байт. И только в случае, когда поразрядная длина числа нечетная, возникают потери 0,5 байт. Таким образом, максимально возможные потери при размещении шестнадцатеричных чисел в памяти не превышают 0,5 байт для каждого числа.

Двоично-шестнадцатеричные преобразования чисел производятся по следующим правилам:

Правило 1. Для того, чтобы преобразовать шестнадцатеричное число в двоичное, следует каждую цифру в составе шестнадцатеричного числа представить в виде двоичного кода на четырех цифровых разрядах.

Правило 2. Для того, чтобы представить двоичное число в виде шестнадцатеричного, его необходимо разбить на группы по 4 двоичных разряда влево и вправо от десятичной точки и каждую группу представить шестнадцатеричной цифрой.

Примечание. Недостающие старшие и младшие разряды двоичного числа дополняются незначащими нулями.

Базовый цифровой ряд шестнадцатеричной системы включает следующие цифры: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

Основанием системы является число $S=16$.

Таблица соответствия десятичных и шестнадцатеричных чисел.

Десятирич. сист.	Шестнадцатерич. сист.
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

Перевод десятичных чисел в шестнадцатеричные выполняется по общим правилам межсистемных преобразований для позиционных чисел. Раздельно для целой и дробной части. При записи шестнадцатеричное число помещается индексом Н. В записи числа в крайней левой позиции всегда должна стоять

цифра. Если записываемое число начинается с шестнадцатеричной цифры в буквенном представлении, перед ней следует ставить 0.

Преобразование шестнадцатеричного числа в десятичное выполняется по правилу вычисления истинного значения позиционного числа.

Сложение и вычитание в дополнительном коде.

При выполнении арифметических операций в современных ЭВМ используется представление положительных чисел в прямом коде (ПК), а отрицательных - в обратном (ОК) или в дополнительном (ДК) кодах. Это можно проиллюстрировать следующей схемой:

ПК	0	Мод.дв.числа		1	1	Мод.дв.числа
ОК	0	Мод.дв.числа		1	1	$\overline{\text{Мод.дв.числа}}$
ДК	0	Мод.дв.числа		1	1	$\text{ОК}+1_{\text{МЗР}}$
	а)				б)	

Представление чисел в ЭВМ:

а) - положительное число; б) - отрицательное число

Общее правило. При алгебраическом сложении двух двоичных чисел, представленных обратным (или дополнительным) кодом, производится арифметическое суммирование этих кодов, включая разряды знаков. При возникновении переноса из разряда знака единица переноса прибавляется к МЗР суммы кодов при использовании ОК и отбрасывается при использовании ДК. В результате получается алгебраическая сумма в обратном (или дополнительном) коде.

Рассмотрим подробнее алгебраическое сложение для случая представления отрицательных чисел в ДК.

При алгебраическом сложении чисел со знаком, результатом также является число со знаком. Суммирование происходит по всем разрядам, включая знаковые, которые при этом рассматриваются как старшие. При возникновении переноса из старшего разряда единица переноса отбрасывается и возможны два варианта результата:

- знаковый разряд равен нулю: результат - положительное число в ПК;
- знаковый разряд равен единице: результат - отрицательное число в ДК.

Для определения абсолютного значения результата, его необходимо инвертировать, затем прибавить единицу.

Пример:

Вычислить алгебраическую сумму $58 - 23$.

$58_{(10)} \rightarrow 0011\ 1010_{(2)}$	- ПК	
$-28_{(10)} \rightarrow 1001\ 0111_{(2)}$	- ПК	} Число отрицательное - необходимо перевести в ДК (быстрый перевод)
$1110\ 1001_{(2)}$	- ДК	

$$\begin{array}{r}
 0011\ 1010 \\
 + \\
 \hline
 1110\ 1001 \\
 \hline
 1 \leftarrow 0010\ 0011_{(2)} \quad (\text{ПК}) = 35_{(10)} \\
 \text{перенос}
 \end{array}$$

Перенос из знакового разряда отбрасываем.
Число является положительным в ПК.

Пример:

Вычислить алгебраическую сумму $26 - 34$.

$$\begin{array}{r}
 26_{(10)} \rightarrow 0001\ 1010_{(2)} \quad - \text{ПК} \\
 -34_{(10)} \rightarrow 1010\ 0010_{(2)} \quad - \text{ПК} \\
 \phantom{-34_{(10)}} 1101\ 1110_{(2)} \quad - \text{ДК} \quad \leftarrow \text{Быстрый перевод}
 \end{array}$$

$$\begin{array}{r}
 0001\ 1010 \\
 + \\
 \hline
 1101\ 1110 \\
 \hline
 0 \leftarrow 1111\ 1000_{(2)} \\
 \text{перенос}
 \end{array}$$

Перенос из знакового разряда отсутствует. Число отрицательное в ДК, так как знаковый разряд равен 1.

Вычислим результат, преобразовав его из ДК в ПК :

$$\begin{array}{r}
 1111\ 1000_{(2)} \quad (\text{ДК}) \\
 1000\ 0111 \quad \text{Инверсия всех разрядов, кроме знакового} \\
 + \\
 \hline
 1000\ 1000_{(2)} \quad (\text{ПК}) = -8_{(10)}
 \end{array}$$

Пример:

Вычислить алгебраическую сумму $-5 - 1$.

$$\begin{array}{l}
 -5_{(10)} \rightarrow 1101 \text{ (ПК)} \rightarrow 1011 \text{ (ДК)} \\
 -1_{(10)} \rightarrow 1001 \text{ (ПК)} \rightarrow 1111 \text{ (ДК)}
 \end{array}$$

$$\begin{array}{r}
 1011 \\
 + \\
 \hline
 1111 \\
 \hline
 1 \leftarrow 1010_{(2)} \\
 \text{перенос}
 \end{array}$$

Перенос из знакового разряда отбросить.
Число отрицательное в ДК, так как знаковый разряд равен 1.

Запишем результат, преобразовав его из ДК в ПК:

$$\begin{array}{r}
 1010_{(2)} \quad (\text{ДК}) \\
 1101 \quad \text{Инверсия значащих разрядов (кроме знакового)} \\
 + \\
 \hline
 1110_{(2)} \quad (\text{ПК}) = -6_{(10)}
 \end{array}$$

Признак переполнения разрядной сетки

При алгебраическом суммировании двух чисел, помещающихся в разрядную сетку, может возникнуть переполнение, то есть образуется сумма,

требующая для своего представления на один двоичный разряд больше, чем разрядная сетка слагаемых. Предполагается, что положительные числа представляются в прямом коде, а отрицательные – в дополнительном.

Признаком переполнения является наличие переноса в знаковый разряд суммы при отсутствии переноса из знакового разряда (положительное переполнение) или наличие переноса из знакового разряда суммы при отсутствии переноса в знаковый разряд (отрицательное переполнение).

При положительном переполнении результат операции положительный, а при отрицательном переполнении - отрицательный.

Если и в знаковый, и из знакового разряда суммы есть переносы или этих переносов нет, то переполнение отсутствует.

Рассмотрим простейшие примеры с трехбитовыми словами. Диапазон чисел, которые они представляют, равен от -4 до +3. В рассматриваемых словах 1 бит знака и 2 информационных бита.

1. Алгебраическое суммирование без переноса.

$$\begin{array}{r}
 2+1=3 \\
 2_{(10)} \rightarrow 010_{(2)} \text{ ПК} \\
 1_{(10)} \rightarrow 001_{(2)} \text{ ПК}
 \end{array}
 \quad
 \begin{array}{r}
 010_{(2)} \\
 + 001_{(2)} \\
 \hline
 0 \leftarrow 011_{(2)} \\
 0 \uparrow \\
 \text{перенос}
 \end{array}
 \quad
 \text{ПК} = 3_{(10)}$$

Так как перенос в знаковый разряд или из знакового разряда суммы отсутствует, то переполнения нет.

Результат - положительное число в ПК, равное 3.

2 Основы теории логического проектирования цифровых устройств

Задачи, решаемые при разработке цифровых логических устройств, можно разделить на две категории:

Синтез - это процесс построения схемы цифрового устройства по заданию.

Анализ - процесс обратный синтезу.

Модель дискретного устройства, отражающая только его свойства по переработке сигналов, называется дискретным (цифровым) автоматом (рисунок 1).

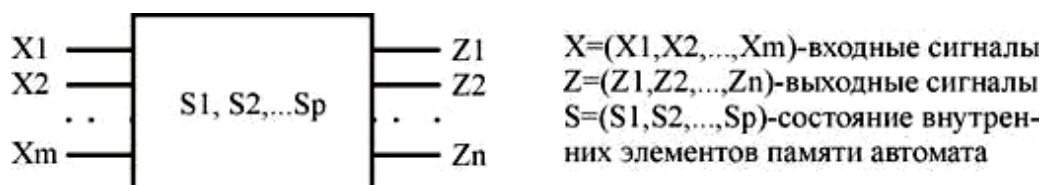


Рисунок 1 – Дискретный цифровой автомат

В общем случае, модель представляет собой многополюсный черный ящик с m входами и n выходами (рис.1). Состояние автомата определяется состояниями сигналов на его входах и выходах. Совокупность входных и

выходных переменных X и Z образуют входное и выходное слово автомата, соответственно.

Различные значения входных переменных образуют алфавит (т.к. алфавит входных и выходных переменных един, в дальнейшем будет рассматриваться только один алфавит). В цифровой технике алфавит входного (выходного) слова содержит два значения (две буквы) "1" и "0".

Каждое слово - набор переменных на входе или на выходе автомата, отличается от другого слова хотя бы одной буквой. Каждая буква слова поставлена в соответствие с номером входа (выхода) автомата.

3 Базовые элементы ЭВМ

Элементами электронных вычислительных машин называются наименьшие функциональные части, на которые можно разделить машину при ее логическом проектировании и технической реализации. Элементы ЭВМ выполняют хранение, преобразование и передачу двоичных переменных, а также ряд вспомогательных функций: задержку сигнала во времени, формирование сигнала с определенными физическими характеристиками и т. п.

Основу элементной базы ЭВМ составляют цифровые интегральные схемы (ИС). Их сложность характеризуется степенью функциональной интеграции $K_{\text{и}}$ ($K_{\text{и}} = \lg N_{\text{эл}}$ где $N_{\text{эл}}$ — число элементов И — НЕ либо ИЛИ — НЕ, расположенных на кристалле ИС).

В зависимости от сложности интегральные схемы подразделяются на следующие типы:

МИС — малые интегральные схемы, содержащие один или несколько логических элементов ($K_{\text{и}} \approx 1$);

СИС — средние интегральные схемы, содержащие один или несколько функциональных узлов (сумматоров, счетчиков и др.) и имеющие обычно $K_{\text{и}} = 1 \dots 2$;

БИС — большие интегральные схемы, содержащие одно или несколько функциональных устройств (арифметическое устройство, арифметико-логическое устройство АЛУ, устройство управления УУ, запоминающее устройство ЗУ) и имеющие $K_{\text{и}} = 2 - 4$;

СБИС — сверхбольшие БИС, выполняющие функции целых цифровых систем (например, микроЭВМ) и имеющие $K_{\text{и}} > 4$.

По типу обрабатываемых сигналов элементы ЭВМ подразделяются на потенциальные, импульсные и импульсно-потенциальные. Длительность потенциального сигнала не ограничена сверху, как правило, кратна длительности такта T и определяется частотой смены информации.

Длительность импульсного сигнала имеет некоторое постоянное стандартное значение $t_{\text{и}} < T$ и но не зависит от частоты смены информации. Типы сигналов определяют три способа представления (кодирования) информации: потенциальный, импульсный и динамический.

По функциональному назначению элементы делятся на логические, элементы памяти и специальные. К логическим относят элементы,

реализующие функции из базисной системы (базисную функцию). Элементы памяти (триггеры) предназначены для запоминания (хранения) информации, а специальные элементы — для физического преобразования электрических сигналов, т. е. усиления, формирования, генерирования, задержки и т. п.

Контрольные вопросы:

1. Назначение АЛУ и их виды.
2. Структура арифметико-логического устройства.
3. Принципы выполнения арифметических операций в ЭВМ над числами с фиксированной запятой.
4. Принципы выполнения арифметических операций в ЭВМ над числами с плавающей запятой.
5. Объясните принципы арифметики двоично-кодированных чисел.
6. Как осуществляется контроль выполнения арифметических операций.
7. Задачи, решаемые при разработке цифровых логических устройств.
8. Что составляет основу элементной базы ЭВМ?
9. Как подразделяются интегральные схемы?

Тема 3. Функциональные узлы ЭВМ

1. Функциональные узлы ЭВМ.
2. Назначение, основные параметры, классификация и принципы построения функциональных узлов ЭВМ.

1 Функциональные узлы ЭВМ

Функциональные узлы ЭВМ представляют отдельные части ЭВМ, реализующие заданные функции в виде определенных машинных операций. Они выполняются, как правило, на основе единых серий элементов ЭВМ. Промышленность выпускает серии элементов (и узлов), на основе которых может быть реализована ЭВМ, выполняющая любые заложенные в нее функции.

Примером подобной серии элементов и узлов может служить комплекс интегральных элементов: К580, КР580, К1810, К1801.

Различают следующие типовые функциональные узлы ЭВМ:

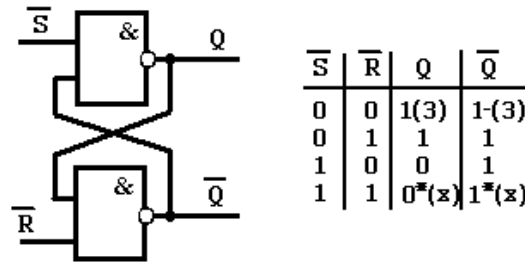
- триггеры;
- регистры;
- счетчики;
- шифраторы;
- дешифраторы;
- мультиплексоры;
- демультимплексоры;
- сдвигатели;
- распределители (уровней и импульсов) и др.

2 Назначение, основные параметры, классификация и принципы построения функциональных узлов ЭВМ

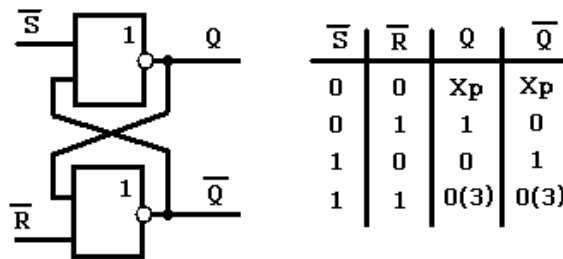
Триггер – это функциональный узел ЭВМ, содержащий элемент памяти и схемы управления данным элементом. Существуют различные классификации триггеров, в частности:

1. По времени поступления сигналов:
 - асинхронные, синхронные.
 - одноктактные - одноступенчатые, двуступенчатые;
 - многотактные.
2. По базису, т.е. используемым логическим элементам:
 - в базисе И–НЕ;
 - в базисе ИЛИ-НЕ;
 - в смешанном базисе.
3. По типу: RS, D, JK, T, DV, DT, RST и др.

Асинхронные триггеры бывают только RS-типа и представляют логическую схему вида:



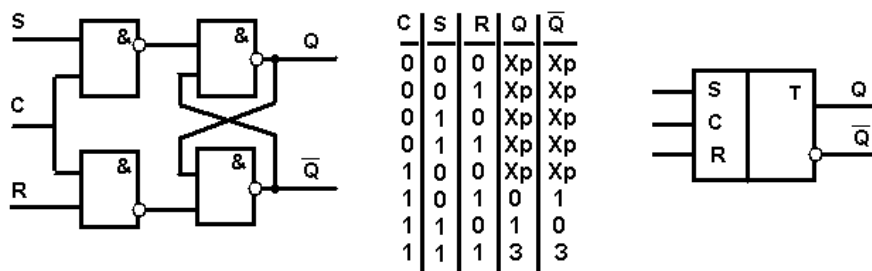
Асинхронный RS-триггер в базисе И-НЕ



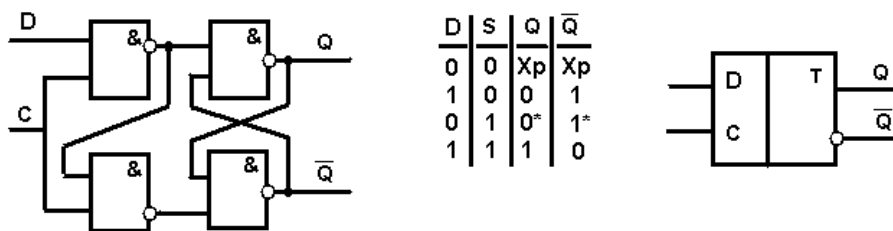
Асинхронный RS-триггер в базисе ИЛИ-НЕ

Синхронные триггеры могут быть более разнообразного типа, в частности, следующие:

Синхронные одноклапчатые одноступенчатые триггеры.

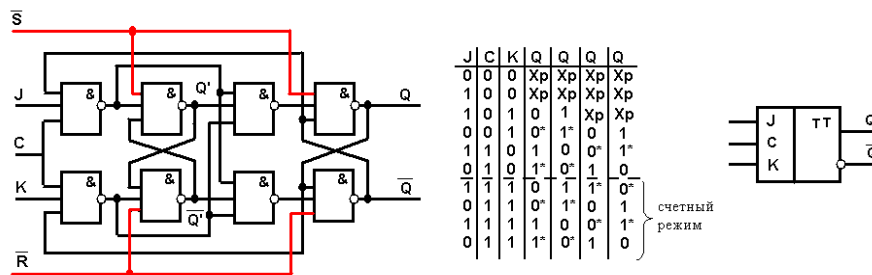


Синхронный RS-триггер в базисе И-НЕ

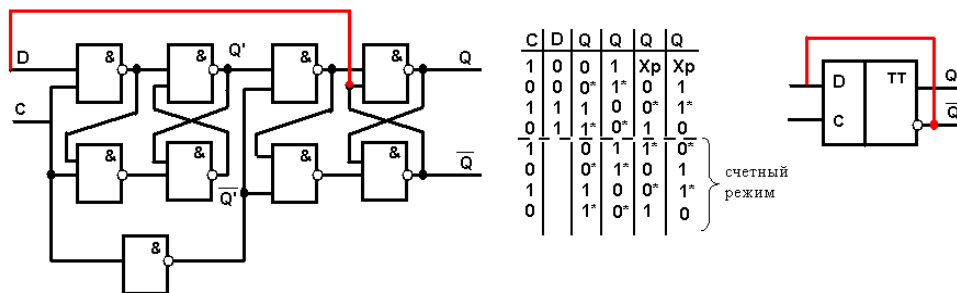


D-триггер в базисе И-НЕ

Синхронные одноктактные двуступенчатые триггеры.



JK-триггер в базисе И-НЕ с запрещающими связями



D-триггер в базисе И-НЕ с инвертором

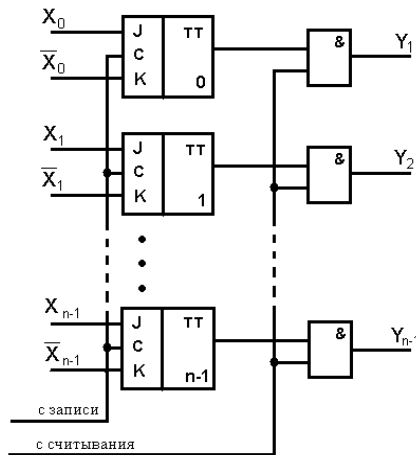
Триггеры, имеющие в названии типа букву V – имеют дополнительный синхровход с названием V. Триггеры RST и DvT выполняют комбинированные функции, RST – может выполнять функции RS и T-триггеров, DvT – имеет дополнительный синхровход V, может выполнять функции D и T-триггеров.

Регистр – это функциональный узел ЭВМ, предназначенный для записи, хранения, считывания n-разрядного слова информации. Регистры строятся на одноктактных или двуктактных двуступенчатых триггерах. Они позволяют осуществлять выполнение логических операций, таких как: \vee, \wedge, \oplus , а также сдвиг информации на заданное количество разрядов.

Типы регистров различаются:

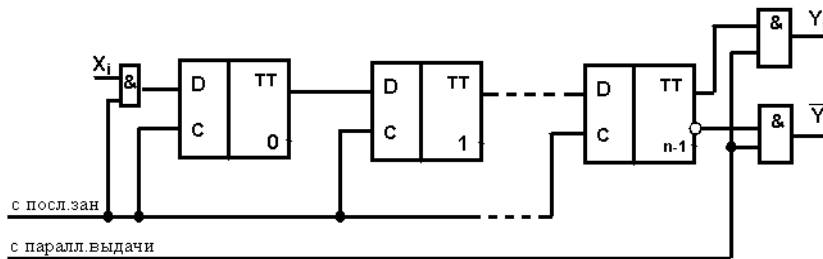
- по способу записи-считывания – на:
 - последовательные – т.е. с последовательной записью и считыванием информации;
 - параллельные – т.е. с параллельной записью и считыванием информации;
 - последовательно-параллельные – с последовательной записью и параллельным считыванием информации;
 - параллельно-последовательные – с параллельной записью и последовательным считыванием информации;
 - универсальные – с параллельной и последовательной записью и параллельным и последовательным считыванием информации.
- по фазности поступающей и выходной информации:
 - однофазные на D-триггерах;

- парафазные на RS и JK-триггерах;
- комбинированные:
 - однофазное занесение, парафазная выдача;
 - парафазное занесение, однофазная выдача.
- по способу тактирования:
 - одноктактные, т.е. управляемые одной последовательностью тактовых сигналов;
 - многотактные, т.е. управляемые несколькими последовательностями тактовых сигналов.



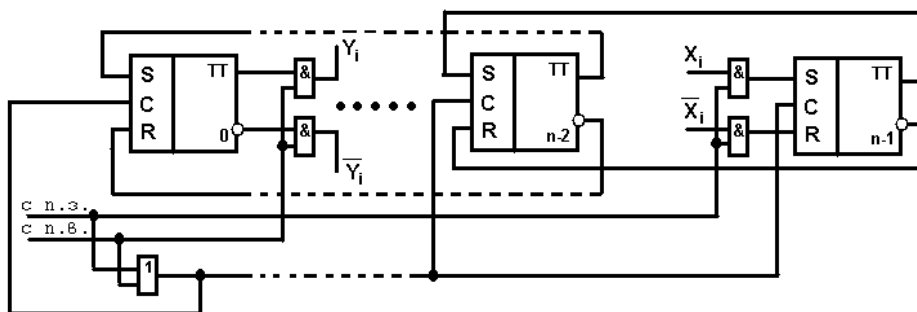
Параллельный регистр на JK-триггерах

Регистр характеризуется парафазным занесением однофазным считыванием информации. Возможна также парафазная выдача информации.



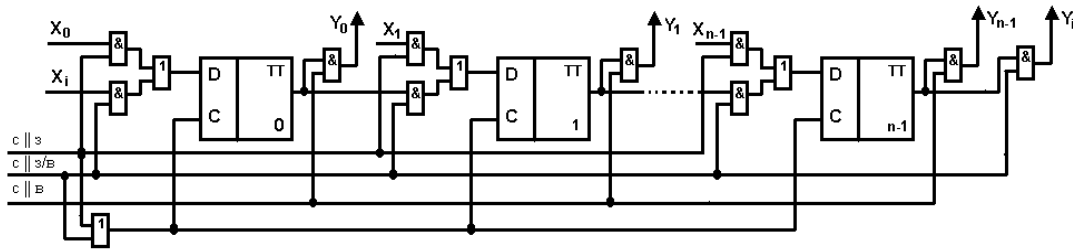
Последовательный регистр на D-триггерах

Регистр характеризуется однофазным занесением информации слева направо и парафазной выдачей информации.



Последовательный регистр на RS-триггерах

В регистре осуществляется парафазное занесение информации справа налево и однофазным считыванием.



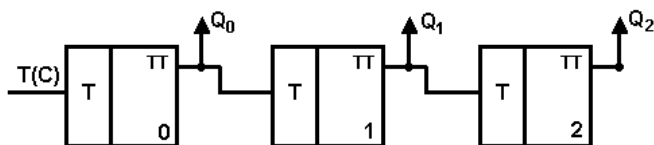
Универсальный регистр на D-триггерах

В регистре реализуется однофазное, последовательное, слева направо и параллельное занесение информации, последовательное и параллельное считывание информации (однофазное).

Счетчик – это функциональный узел ЭВМ, предназначенный для подсчета количества поступивших на его вход сигналов и выдачи его в виде n -разрядного двоичного кода. Максимальное число, до которого может считать счетчик, образует коэффициент пересчета $N=2^n$, где N – разрядность счетчика.

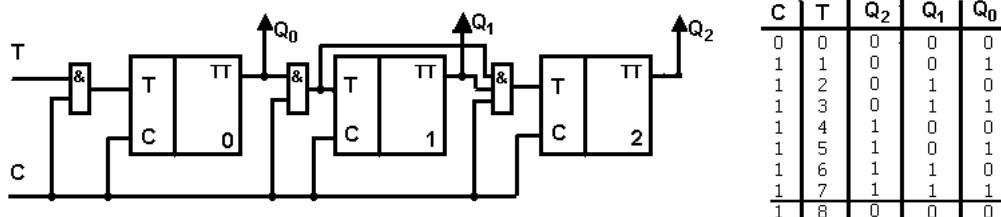
Счетчики строятся на одноктактных двуступенчатых Т-триггерах. Они классифицируются по следующим признакам:

1. По направлению счета:
 - суммирующие;
 - вычитающие;
 - реверсивные;
2. По основанию системы счисления:
 - двоичные;
 - недвоичные:
 - а) двоично-десятичные;
 - б) с произвольным коэффициентом;
3. по времени поступления сигналов:
 - асинхронные;
 - синхронные;
4. по способу организации разрядного переноса:
 - с последовательным переносом;
 - с параллельным переносом;
 - с комбинированным (групповым) переносом.

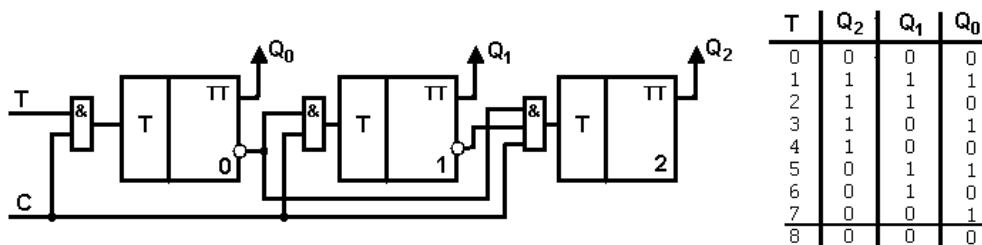


T(C)	Q ₂	Q ₁	Q ₀
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

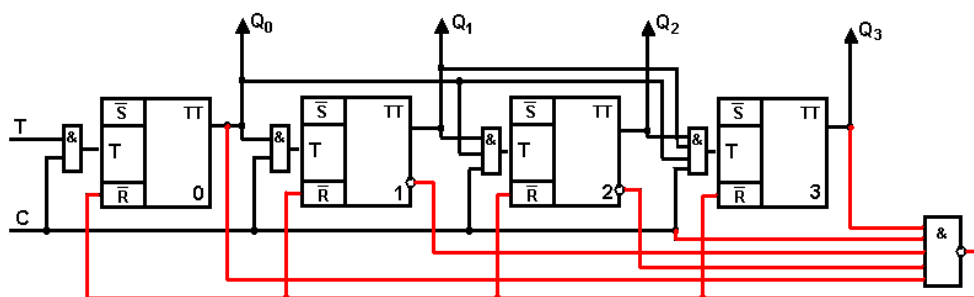
Асинхронный суммирующий двоичный счетчик с последовательным переносом



Синхронный суммирующий двоичный счетчик с параллельным переносом. Переключение (подсчет входных сигналов) возможно только в присутствии синхросигнала С.



Вычитающий синхронный двоичный счетчик с параллельным переносом



Десятичный синхронный суммирующий счетчик с параллельным переносом.

Дешифратор – это устройство, предназначенное для преобразования двоичного кода в напряжение логической единицы (логического нуля) на том выходе, номер которого совпадает со значением двоичного кода на входе. При n входах в *полном дешифраторе* имеется 2^n выходов, т.е. для каждой комбинации входных сигналов имеется соответствующий выход. Дешифратор, у которого при n входах число выходов меньше 2^n , называется *неполным*. Другое название дешифратора - *декодер*. Принцип работы полного трехразрядного дешифратора рассмотрим на примере его таблицы истинности.

Входы			Выходы							
X_3	X_2	X_1	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Соответствующие таблице истинности ФАЛ имеют вид

$$Y_0 = \overline{X_3} \cdot \overline{X_2} \cdot \overline{X_1}$$

$$Y_1 = \overline{X_3} \cdot \overline{X_2} \cdot X_1$$

$$Y_2 = \overline{X_3} \cdot X_2 \cdot \overline{X_1}$$

$$Y_3 = \overline{X_3} \cdot X_2 \cdot X_1$$

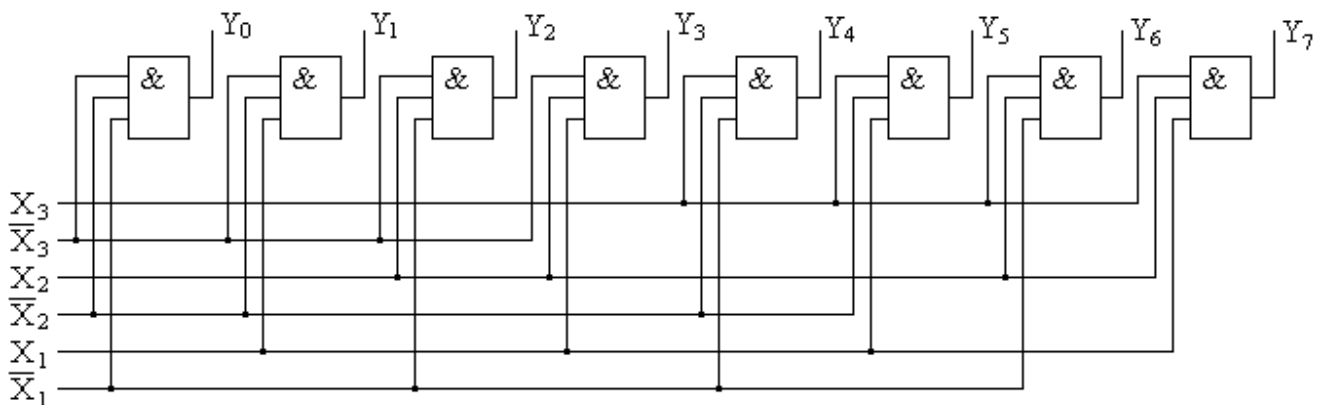
$$Y_4 = X_3 \cdot \overline{X_2} \cdot \overline{X_1}$$

$$Y_5 = X_3 \cdot \overline{X_2} \cdot X_1$$

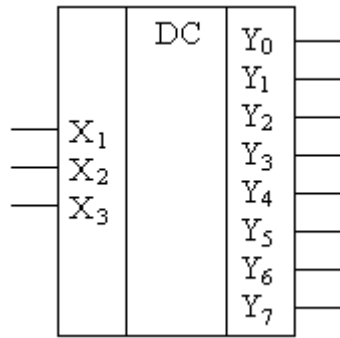
$$Y_6 = X_3 \cdot X_2 \cdot \overline{X_1}$$

$$Y_7 = X_3 \cdot X_2 \cdot X_1$$

Структурная схема трехразрядного дешифратора, синтезированная на основании полученных ФАЛ приведена на рис. 4.1,а, а его УГО - на рис. 4.1,б.



а)



б)

Рисунок 1 - Структурная схема и УГО трехразрядного дешифратора.

В общем случае логические уравнения для выходных переменных дешифратора n -разрядного числа имеют вид

$$Y_0 = \overline{X_n} \cdot \overline{X_{n-1}} \cdot \dots \cdot \overline{X_2} \cdot \overline{X_1}$$

$$Y_1 = \overline{X_n} \cdot \overline{X_{n-1}} \cdot \dots \cdot \overline{X_2} \cdot X_1$$

$$Y_2 = \overline{X_n} \cdot \overline{X_{n-1}} \cdot \dots \cdot X_2 \cdot \overline{X_1}$$

$$Y_3 = \overline{X_n} \cdot \overline{X_{n-1}} \cdot \dots \cdot X_2 \cdot X_1$$

.....

$$Y_{2^n-1} = X_n \cdot X_{n-1} \cdot \dots \cdot X_2 \cdot X_1$$

Построенные по полученным формулам дешифраторы называются *линейными*. К преимуществу линейных дешифраторов можно отнести высокое быстродействие, поскольку входные переменные одновременно поступают на все элементы И. Одновременно, без дополнительных задержек, формируется и результат на выходах этих элементов. Очевидно, что для реализации линейного дешифратора n -разрядного числа необходимо иметь 2^n логических элементов И с n -входами. В существующих микросхемах логических элементов количество входов ограничено. Следовательно, ограничена и разрядность реализуемых на их основе линейных дешифраторов, что является недостатком. Кроме того, недостатком является и то, что предыдущие элементы, работающие на входы дешифратора, должны иметь высокую нагрузочную способность, т.е. должны быть рассчитаны на подключение большого числа логических элементов И. Каждый из входов дешифратора подключен к $0,5 \cdot 2^n$ логическим элементам И. Поскольку нагрузочная способность базовых логических элементов ИС не превышает величины $N=10,20$, то максимальная разрядность дешифрируемых чисел для линейных дешифраторов $n=4,5$.

Указанного недостатка лишены *пирамидальные дешифраторы*. Принцип построения этих дешифраторов состоит в том, что сначала строят линейный дешифратор для двухразрядного числа X_1, X_2 , для чего необходимы $2^2=4$ двухвходовые схемы И. Далее, каждая полученная конъюнкция логически умножается на входную переменную X_3 в прямой и инверсной форме.

Полученная конъюнкция снова умножается на входную переменную X_4 в прямой и инверсной форме и т.д. Нарастивая таким образом структуру, можно построить пирамидальный дешифратор на произвольное число входов. На рис. 2 приведена структура пирамидального дешифратора для трех разрядов.

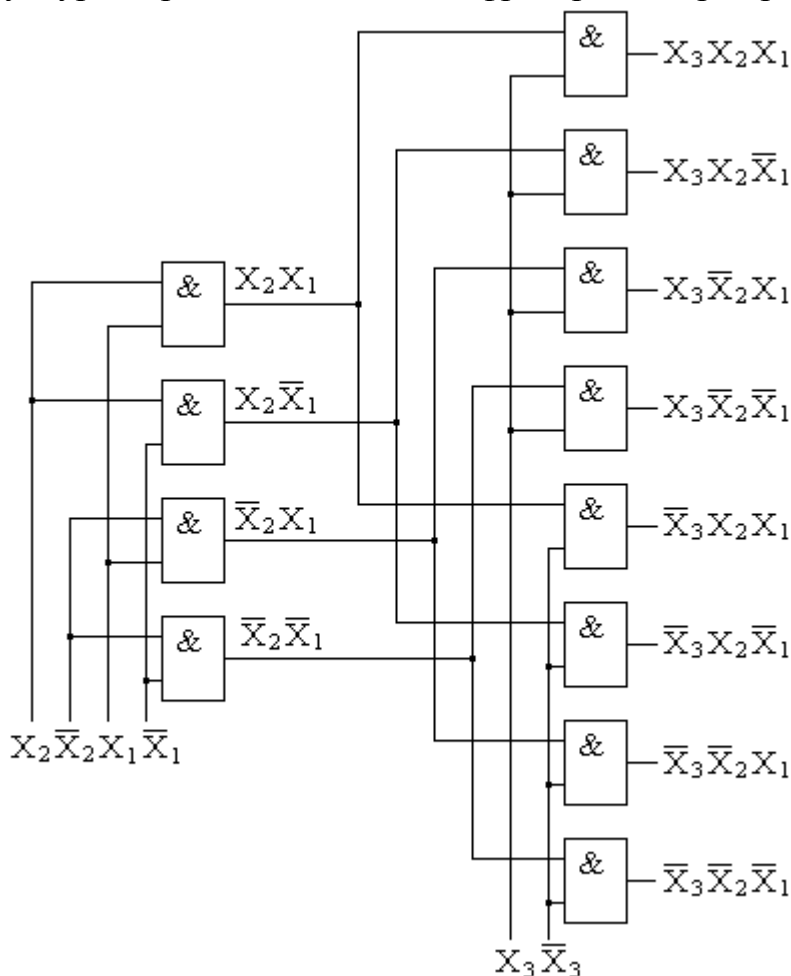


Рисунок 2 - Пирамидальный дешифратор для трехразрядного числа.

Характерным отличием пирамидальных дешифраторов от линейных является использование только двухвходовых логических элементов вне зависимости от разрядности дешифрируемого числа. В то же время количество логических элементов в пирамидальном дешифраторе больше. Однако следует иметь в виду, что количество логических элементов, располагаемых в одном корпусе ИС, определяется главным образом требуемым количеством выводов. Следовательно, в одном корпусе ИС можно расположить большее число двухвходовых элементов, чем трехвходовых, четырехвходовых и т.д. И значит, пирамидальная структура дешифратора по числу корпусов ИС может оказаться более предпочтительной, чем линейная.

Шифраторы выполняют задачу обратную той, которую выполняют дешифраторы: появление логической единицы (логического нуля) на определенном входе приводит к появлению соответствующей кодовой комбинации на выходе. Также как и дешифраторы, шифраторы бывают полными и неполными. Работа восьмивходового полного шифратора задается следующей таблицей истинности:

Входы								Выходы		
X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0	Y_3	Y_2	Y_1
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

На основании таблицы истинности можно записать ФАЛ, задающие работу восьмивходового шифратора:

$$Y_1 = X_1 \vee X_3 \vee X_5 \vee X_7$$

$$Y_2 = X_2 \vee X_3 \vee X_6 \vee X_7$$

$$Y_3 = X_4 \vee X_5 \vee X_6 \vee X_7$$

Синтезированная на основании приведенных логических уравнений структурная схема шифратора представлена на рис. 4.3,а, а его условное графическое обозначение – на рис. 4.3,б.

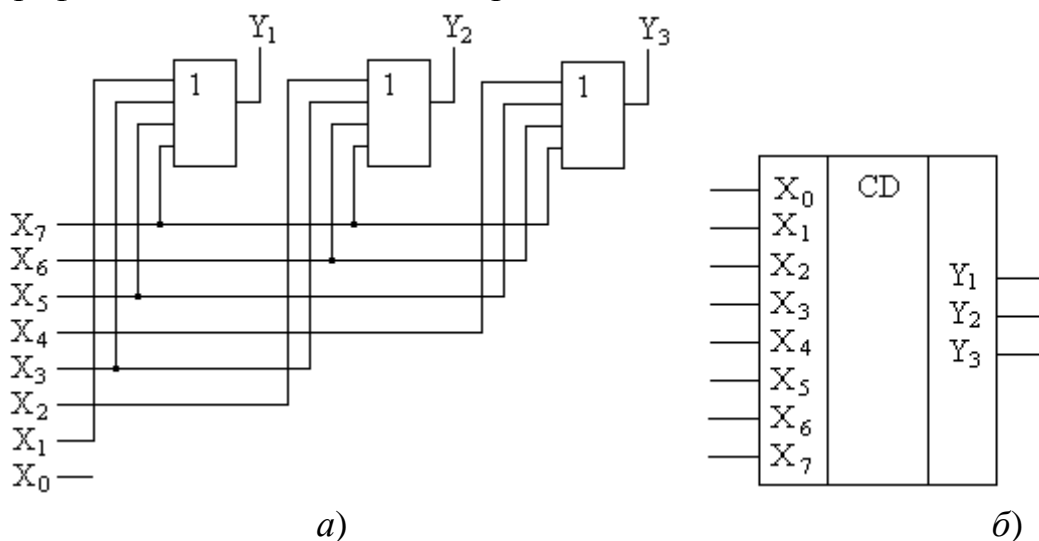


Рис. 3. Структура и УГО восьмивходового шифратора.

Мультиплексор - комбинационное цифровое устройство, которое обеспечивает передачу на единственный выход F одного из нескольких входных сигналов D_j в соответствии с поступающим адресным кодом A_i . При наличии n адресных входов можно реализовать $M=2^n$ комбинаций адресных сигналов, каждая из которых обеспечивает выбор одного из M входов. Чаще всего используются мультиплексоры «из 4 в 1» ($n=2, M=4$), «из 8 в 1» ($n=3, M=8$), «из 16 в 1» ($n=4, M=16$). Правило работы мультиплексора «из 4 в 1» можно задать таблицей истинности:

Входы		Выход
A_1	A_0	F
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

Логическое выражение для выходной функции, заданной таблицей, можно записать в виде

$$F = D_0 \bar{A}_1 \bar{A}_0 \vee D_1 \bar{A}_1 A_0 \vee D_2 A_1 \bar{A}_0 \vee D_3 A_1 A_0.$$

В соответствии с полученной формулой для реализации мультиплексора можно использовать логические элементы И, ИЛИ, НЕ. Синтезированная структурная схема мультиплексора показана на рисунке 4 а, а его условное графическое обозначение – на рисунке 4 б.

Мультиплексирование при большом числе входов можно выполнить пирамидальным каскадированием мультиплексоров, как это показано на рисунке 5. На рисунке показано каскадирование мультиплексоров «из 4 в 1» для реализации функции мультиплексирования «из 16 в 1».

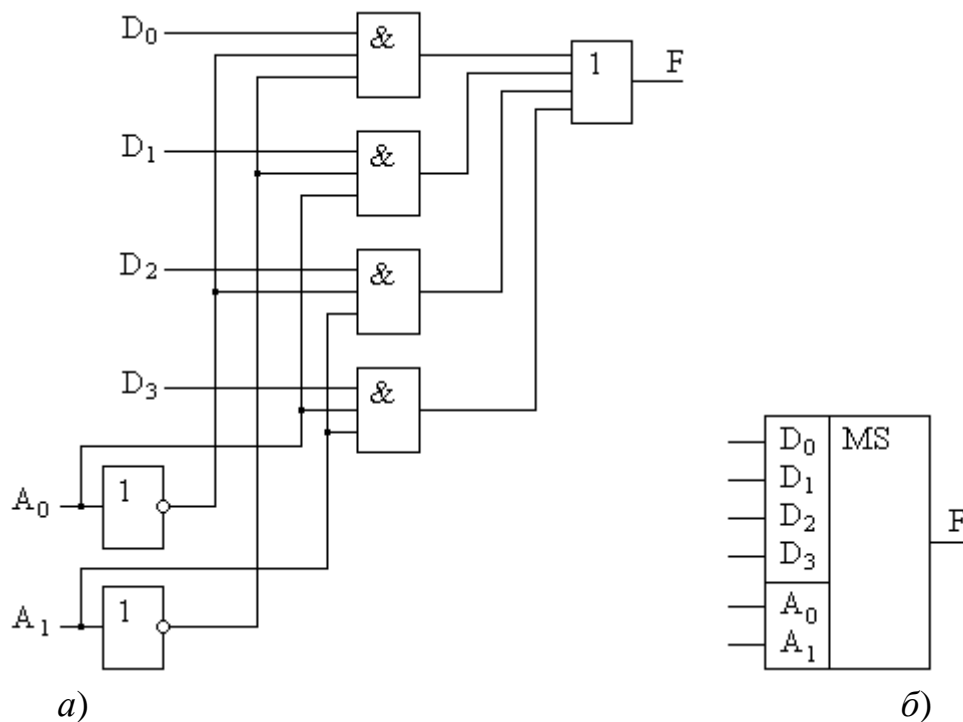


Рисунок 4. Структура и УГО мультиплексора «из 4 в 1»

Мультиплексоры первого уровня управляются адресными сигналами A_0 и A_1 , а мультиплексоры второго – адресными сигналами A_2 и A_3 . Каждый из мультиплексоров первого уровня выбирает один из четырех разрядов D_j . Первый мультиплексор выбирает один из разрядов $D_0 - D_3$, второй мультиплексор – один из разрядов $D_4 - D_7$ и т.д. Выходы с мультиплексоров первого уровня объединяются в мультиплексоре второго уровня, который осуществляет окончательную коммутацию и формирование выходного сигнала F .

Демультимплексор выполняет функцию, обратную мультиплексору, т.е. в соответствии с принятой адресацией A_i направляет информацию с единственного входа D на один из M выходов F_j . При этом на остальных выходах будут логические нули (единицы). Принцип работы демультимплексора «из 1 в 4» иллюстрируется таблицей истинности:

Входы		Выходы			
A_1	A_0	F_3	F_2	F_1	F_0
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	D	0	0
1	1	D	0	0	0

Логические выражения для каждого из выходов можно представить в виде:

$$F_0 = D\bar{A}_1\bar{A}_0$$

$$F_1 = D\bar{A}_1A_0$$

$$F_2 = DA_1\bar{A}_0$$

$$F_3 = DA_1A_0$$

Структурная схема, реализующая демультимплексор «из 1 в 4» приведена на рисунке 6 а, а его условное графическое обозначение – на рисунке 6 б.

Как и в случае мультиплексора, схему демультимплексора можно реализовать с помощью дешифратора. Действительно, ФАЛ демультимплексора отличается от ФАЛ дешифратора только наличием входного сигнала D в конъюнкциях с адресными входами. Следовательно, объединив выходы дешифратора с входом D с помощью стробирующих элементов И, можно получить демультимплексор. Мультиплексоры и демультимплексоры часто называют еще цифровыми коммутаторами.

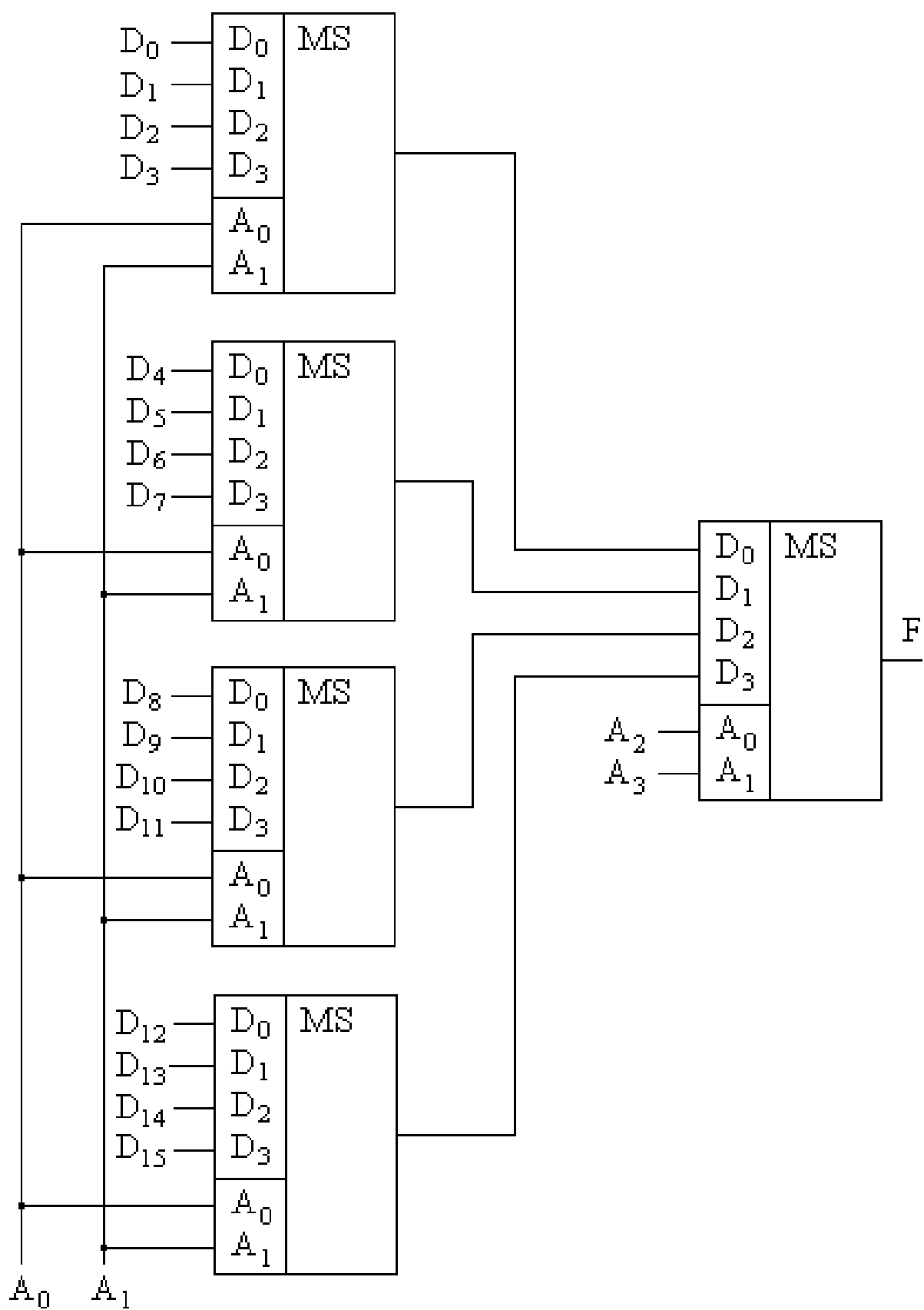


Рисунок 5 - Пирамидальное каскадирование мультиплексоров.

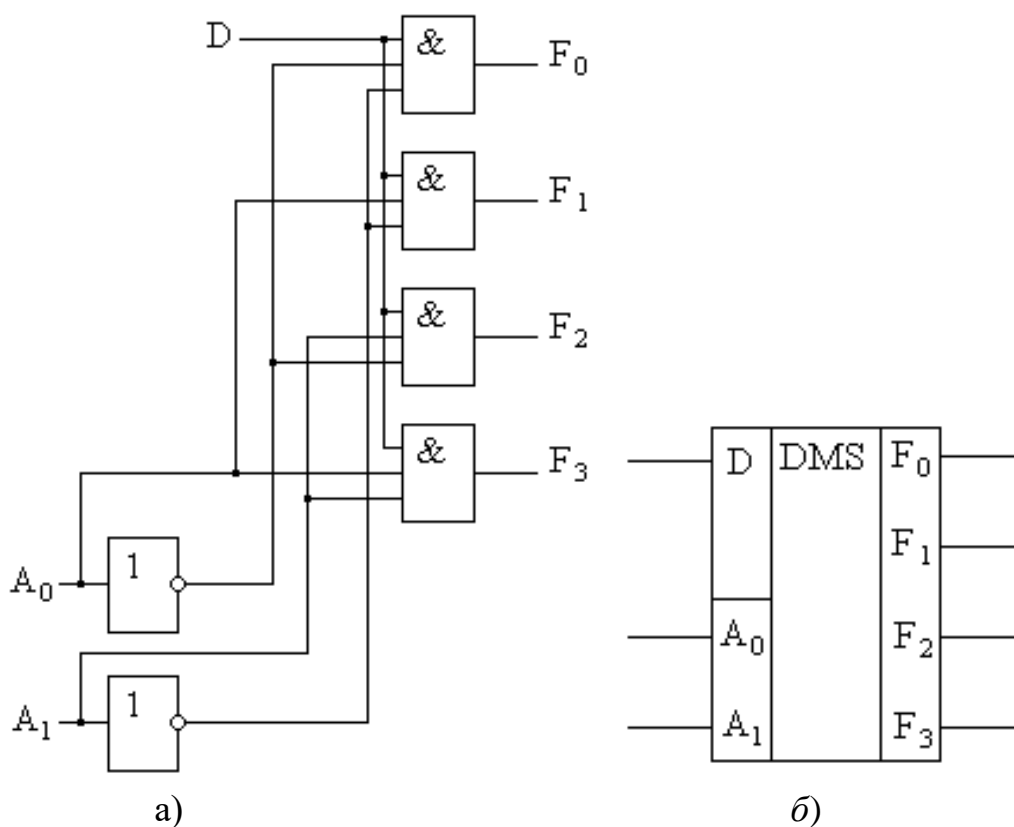


Рисунок 6 - Структурная схема и УГО демультиплексора «из 1 в 4».

Контрольные вопросы

1. Перечислите основные функциональные узлы ЭВМ.
2. Какие логические устройства называют последовательными?
3. Объясните схему триггерной ячейки.
4. Назовите наиболее распространенные типы триггеров.
5. Каковы особенности D-триггера? Что называют цифровыми регистрами?
6. Какие существуют виды регистров, каковы их особенности?
7. К какой группе логических устройств относят счетчики?
8. Поясните классификацию счетчиков.
9. В каких режимах может работать счетчик?
10. Назначение и принципы работы двоично-кодированных счетчиков.
11. Назначение шифратора и дешифратора.
12. Какие вы знаете виды шифраторов?
13. Какие устройства называют мультиплексором и демультиплексором?
14. Как еще их называют?
15. Для чего используется пирамидальное каскадирование мультиплексоров?
16. Поясните особенности реализации мультиплексора на базе дешифратора.

Тема 4. Запоминающие устройства (ЗУ) ЭВМ

1. Определение, назначение, основные характеристики и классификация запоминающих устройств (ЗУ) ЭВМ.
2. Способы организации памяти.
3. Способы повышения емкости памяти.

1 Определение, назначение, основные характеристики и классификация запоминающих устройств (ЗУ) ЭВМ

Информация в ПК хранится в электронной памяти. Память- это устройство, служащее для хранения данных в организованном порядке. Различают несколько видов электронной памяти - оперативную, постоянную, внешнюю и эргонезависимую.

Оперативная память (ОЗУ) или RAM (Random Access Memory) относится к одному из важнейших компонентов ЭВМ. От ее параметров в значительной степени зависят функциональные возможности всей системы. Она позволяет не только считывать хранящиеся в ней данные, но и записывать новые. В микросхемах оперативной памяти хранится информация, которая меняется в процессе работы ПК. RAM отличается высокой скоростью записи и чтения. При выключении ПК данные, хранящиеся в RAM, теряются. Объем оперативной памяти зависит от количества и типа используемых микросхем.

Характеристики оперативной памяти и особенности ее устройства являются важнейшим фактором, от которого зависит быстродействие компьютера. Даже при наличии «быстрого» процессора скорость выборки данных из памяти может оказаться невысокой, и именно эта невысокая скорость работы ОЗУ будет определять быстродействие ЭВМ. Время цикла работы с памятью t_m особенно заметно больше чем время цикла работы центрального процессора t_c . При обращении к ОЗУ, время обращения $T=t_m+t_c$, в период этого времени ни одно устройство не может работать. Таким образом, возникает проблема доступа к ОП.

Оперативная память выполнена обычно на микросхемах динамического типа с произвольной выборкой DRAM (Dynamic Random Access Memory). Каждый бит такой памяти представляется в виде наличия (или отсутствия) заряда на конденсаторе, образованном в структуре полупроводникового кристалла. В современных ПК DRAM используется обычно в качестве оперативной памяти общего назначения, а также как память видеоадаптера. Из применяемых типов динамической памяти наиболее известны: DRAM, FPM DRAM, EDO DRAM, BEDO DRAM, ED RAM, CD RAM, Synchronous DRAM, DDR SDRAM и SL RAM; видеопамять: MDRAM, VRAM, WRAM, SGRAM, RDRAM и др.

Другой, более дорогой тип памяти – статический. SRAM (Static RAM) в качестве элементарной ячейки использует так называемый статический триггер, схема которого состоит из нескольких транзисторов. Статический тип памяти обладает более высоким быстродействием. SRAM обычно применяется в

качестве КЭШ-памяти второго уровня для кэширования основного объема ОЗУ. Такая память выполняется на основе ТП, КМОП или БиКМОП – микросхем. По способу доступа к данным может быть как асинхронной, так и синхронной.

Асинхронным называется доступ к данным, который можно осуществлять в произвольный момент времени. SRAM данного типа применялась на материнских платах третьего ÷ пятого поколения процессоров. Время доступа к ячейкам такой памяти составляло от 15нс (33МГц) до 8нс (66МГц).

Для описания характеристик быстродействия ОЗУ применяются так называемые циклы чтения/записи. При обращении к памяти на считывание или запись первого машинного слова расходуется больше тактов, чем на обращение к трем последующим словам.

Для асинхронной SRAM чтение одного слова выполняется за три такта, запись за четыре такта. Чтение нескольких слов определяется последовательностью: 3 – 2 – 2 – 2, а запись: 4 – 3 – 3 – 3.

Синхронная SRAM обеспечивает доступ к данным не в произвольные моменты времени, а синхронно с тактовыми импульсами. В промежутках между ними память может готовить для доступа следующую порцию данных. В большинстве материнских плат пятого поколения используется разновидность синхронной памяти – синхронно-конвейерная SRAM (Pipelined- Burst SRAM). Для которой типичное время одной операции чтения/записи составляет три такта, а групповая операция занимает 3 – 1 – 1 – 1 такта при первом обращении и 1 – 1 – 1 – 1 при последующих обращениях, что обеспечивает ускорение доступа более чем на 30%.

Постоянные запоминающие устройства (ПЗУ) являются обязательным устройством микропрограммного управления ЭВМ. ПЗУ предназначены в первую очередь для организации первоначальной работы ЭВМ при включении питания.

Основным ПЗУ современных ЭВМ является SETUP. В ранних модификациях ЭВМ ПЗУ выполнялось на заводе-изготовителе, и не могла изменять своих данных в процессе функционирования. По мере развития ЭВМ появились перепрограммируемые ПЗУ-ППЗУ. ППЗУ позволяет изменять свои данные, но не в процессе работы ЭВМ, а вне рабочее время, на специальном оборудовании. Для независимости от работы остальных устройств ЭВМ ППЗУ-SETUP снабжены самостоятельным источником питания. В некоторых ЭВМ – Sun может быть поставлена защита от изменения в процессе работы, что повышает надежность защиты от вирусов. В современных ЭВМ SETUP размещается в ППЗУ.

Для уменьшения объема не перепрограммируемых ПЗУ используется организация их на программных логических матрицах – ПЛМ, которые строятся на основе минимизированных выходных функций.

Кроме SETUP ПЗУ может иметь практически любое из устройств ЭВМ. Наличие локального ПЗУ ускоряет работу блоков местного управления (БМУ), что позволяет освободить в большой степени процессор от управления другими устройствами.

Одним из наиболее распространенных видов постоянной памяти относится ROM (Read Only Memory). **ROM** - *память только для чтения*. Русский эквивалент - ПЗУ (Постоянно Запоминающее Устройство).

Если быть совсем точным, данный вид памяти называется *Mask-ROM* (Масочные ПЗУ). ROM устроена в виде адресуемого массива ячеек (матрицы), каждая ячейка которого может кодировать единицу информации. Данные на ROM записывались во время производства путём нанесения по маске (отсюда и название) алюминиевых соединительных дорожек литографическим способом. Наличие или отсутствие в соответствующем месте такой дорожки кодировало "0" или "1". Mask-ROM отличается сложностью модификации содержимого (только путем изготовления новых микросхем), а также длительностью производственного цикла (4-8 недель). Поэтому, а также в связи с тем, что современное программное обеспечение зачастую имеет много недоработок и часто требует обновления, данный тип памяти не получил широкого распространения.

Преимущества данного вида памяти обуславливают:

1. Низкая стоимость готовой запрограммированной микросхемы (при больших объёмах производства).
2. Высокая скорость доступа к ячейке памяти.
3. Высокая надёжность готовой микросхемы и устойчивость к электромагнитным полям.

Недостатки:

1. Невозможность записывать и модифицировать данные после изготовления.
2. Сложный производственный цикл.

Внешняя память располагается на магнитных или оптических дисках. Запись и считывание информации при работе с внешней памятью происходит медленнее, чем при работе с RAM, зато внешняя память имеет большой объем и содержимое ее не меняется при выключении ПК.

Энергонезависимая память представлена микросхемой памяти, в которую записана информация о типе аппаратуры компьютера и его настройке. Настройка ПК может меняться по желанию пользователя, поэтому энергонезависимая память позволяет не только считывать из нее данные, но и записывать. По существу, здесь используется обычная микросхема RAM, но изготовленная по особой CMOS-технологии, обеспечивающей малое потребление энергии при работе этого устройства. Именно поэтому энергонезависимую память часто называют CMOS-памятью. По CMOS-технологии изготавливают все микросхемы для портативных ПК, чтобы обеспечить длительную работу их батарей питания. Микросхема энергонезависимой памяти подключается к батарейке, что сохраняет записанные в микросхеме данные при выключении ПК из сети. В случае, если батарейка истощится, данные о настройке ПК будут уничтожаться при выключении компьютера, и при каждом новом включении придется настраивать конфигурацию ПК заново.

Регистровая память. *Регистровая память*, или СОЗУ (сверх ОЗУ), обладает наибольшим быстродействием. Она обычно реализуется на регистрах, поэтому в МП СОЗУ называется РОН. Объем памяти СОЗУ очень мал.

Каждый из регистров имеет уникальную природу и предоставляет определенные возможности, которые другими регистрами или ячейками памяти не поддерживаются.

Регистры общего назначения процессора используются в операциях большинства инструкций в качестве источника или приемника при перемещении данных и вычислениях, указателей на ячейки памяти и счетчиков. Они могут использоваться для хранения значения данных, в арифметических и логических операциях. Между регистром и памятью может выполняться обмен (запись из регистра в память и наоборот).

Кэширование памяти. Производительность ПК зависит не только от процессора. Большой потенциал производительности ПК связан с организацией и способом функционирования памяти. Важную роль при этом играет кэш-память. Она выполняет функцию буфера между процессором и оперативной памятью.

Буферная память включает запоминающие устройства предназначенные для ускорения работы устройств, тактовая частота которых меньше тактовой частоты самого быстрого устройства ЭВМ – процессора. Буфер также используется для временного хранения информации.

Среди буферных запоминающих устройств наибольший объем, и самое высокое значение занимает КЭШ-память. В современных ЭВМ КЭШ-память имеет неоднородную структуру. Различают КЭШ-память уровней L1 и L2. Уровень L1 - интегрирован в микросхему процессора, имеет объем ~ 10г Кбайт и высокое быстродействие. Кэш первого уровня работает на полной внутренней тактовой частоте процессора. Он встроен во все процессоры класса 486 и выше. Чтобы повысить производительность, для данных и команд часто используется отдельный кэш (так называемая Гарвардская архитектура – противоположность Принстонской, использующей общую память для команд и данных).

Кэш уровня L2 – выпускается в отдельных корпусах (SRAM) и устанавливается на материнской плате (плате программной поддержки работы процессора). Такой кэш обычно работает на внешней тактовой частоте процессора. Для процессоров 486 и Pentium является внешним (устанавливается на системной плате), а у P6 и Pentium 4 располагается в одной упаковке с ядром и подключается к специальной внутренней шине процессора, благодаря чему обеспечивается работа на полной тактовой частоте процессора.

Различают КЭШ-память двух видов - КЭШ-память команд и КЭШ-память данных. В многопроцессорных системах может быть несколько КЭШей одного и того же уровня и назначения.

Данные, которые процессор уже получил из оперативной памяти, остаются в быстрой кэш-памяти, несмотря на то, что они уже обработаны. Подразумевается, что при обмене данными и при выполнении большого числа

операций процессор будет часто запрашивать одни и те же данные и команды. При обращении микропроцессора к памяти сначала производится поиск нужных данных в кэш-памяти.

В случае отсутствия кэш-памяти данные каждый раз считываются из относительно медленной оперативной памяти компьютера. Процессор прерывает свою текущую работу и ждет, пока будут получены необходимые данные для дальнейшей обработки.

Когда кэш-память имеется, данные находятся в специально предназначенной для процессора исключительно быстрой памяти, и при их запросе циклы ожидания отсутствуют.

При управлении кэш-памятью речь идет не о простой памяти, организованной по принципу стека, которая теряет старые данные, если в нее считываются новые. С помощью специальных алгоритмов, наиболее простым из которых является алгоритм LRU (Least Recently Used – удаление наиболее давно использовавшихся данных), чаще используемые данные хранятся в этой памяти дольше, чем другие. Благодаря этому необходимость доступа к основной памяти сводится к минимуму, и компьютер в целом работает быстрее.

Архитектура современных 32-разрядных процессоров включает ряд средств кэширования памяти: два уровня кэша инструкций и данных (L1 Cache и L2 Cache), буферы ассоциативной трансляции (TLB) блока страничной переадресации и буферы записи. Эти средства в разных вариациях (на кристалле, картридже процессора или на системной плате) представлены в системах с процессорами 486, Pentium, P6 и Pentium 4.

Процессор Pentium III имеет L2-кэш объемом 256 Кбайт, который работает на тактовой частоте, равной тактовой частоте ядра. У процессора Pentium 4 L2-кэш имеет объем 256 Кбайт, работает на тактовой частоте ядра, обмен данными с ядром процессора идет по 256-битной шине.

Кэш-контроллер должен обеспечивать когерентность (coherency) – согласованность данных кэш-памяти обоих уровней с данными в основной памяти при том условии, что обращение к этим данным может производиться не только процессором, но и другими активными (bus master) адаптерами, подключенными к шинам (PCI, VLB, ISA и т. д.). Следует также учесть, что процессоров может быть несколько, и у каждого может быть свой внутренний кэш.

Внешняя память ЭВМ организуется в виде накопителей информации. Основными элементами таких накопителей являются носители информации и устройство записи-считывания данных. Носители информации обычно выполняются в виде жёстких дисков, гибких дисков, cd и dvd дисков, магнитных лент.

Первая модель жёсткого диска имела 30 дорожек по 30 секторов, что совпало с калибром ружья “Винчестер” и определило его последующее наименование. Современные жесткие диски используют модули подключения: IDE, MiniIDE, SATA I, SATA II. IDE и MiniIDE. Модуль MiniIDE используется в ноутбуках, поэтому он намного компактнее и дороже, чем его аналог IDE.

На современном рынке появились новые винчестеры, оснащенные модулями SATA I и SATA II. Эти винчестеры стоят в несколько раз дороже, но имеют большую скорость записи. С помощью пары таких винчестеров можно создать RAID массив. Различают следующие типы таких массивов:

1. Raid 0 (striping) объединяет два диска в одну метку тома, тем самым увеличивает скорость записи и объем.

2. Raid 1 (data mirroring) делает точную копию одного винчестера на другом, тем самым увеличивается безопасность сохранения информации.

3. Raid 0+1 объединяет в себе Raid 1 и Raid 0, но требует 4 винчестера.

4. JBOD позволяет установить операционную систему на SATA диск (ни в одном другом режиме массив RAID не позволяет устанавливать операционную систему на диск SATA).

Примерно такие же массивы существуют в SATA II.

Винчестер может иметь собственную кэш – память. Во время запуска программы с винчестера, она заносится в кэш-память и при повторном запуске уже не требует поиска на винчестере.

Классификация накопителей. Для хранения программ и данных в IBM PC-совместимых персональных компьютерах используют различного рода накопители, общая емкость которых, как правило, в сотни раз превосходит емкость оперативной памяти. По отношению к компьютеру накопители могут быть *внешними* и *встраиваемыми* (внутренними). В первом случае такие устройства имеют собственный корпус и источник питания, что экономит пространство внутри корпуса компьютера и уменьшает нагрузку на его блок питания. Встраиваемые накопители крепятся в специальных монтажных отсеках (drive bays) и позволяют создавать компактные системы, которые совмещают в системном блоке все необходимые устройства.

Сам накопитель можно рассматривать как совокупность носителя и соответствующего привода. В связи с этим различают накопители со сменным и несменным носителями. В зависимости от типа носителя все накопители можно подразделить на накопители на магнитной ленте и дисковые накопители. Накопители на магнитной ленте в свою очередь бывают двух видов: накопители на полудюймовых девятидорожечных лентах, работающие в старт-стопном режиме, и стриммеры, работающие в потоковом (инерционном) режиме. Накопители на магнитной ленте называют также устройствами последовательного доступа, так как обратиться к удаленным фрагментам данных можно только после считывания менее удаленных. Накопители же на дисках, как правило, являются устройствами произвольного доступа, поскольку нужные данные могут быть получены без обязательного прочтения им предшествующих.

2 Способы организации памяти

По способу записи / чтения информации на носитель дисковые накопители можно подразделить на магнитные, оптические и магнитооптические. Среди дисковых накопителей можно выделить:

- накопители на флоппи-дисках;
- накопители на несменных жестких дисках (винчестеры);
- накопители на сменных жестких дисках;
- накопители на сменных гибких дисках, использующие эффект Бернулли;
- накопители на магнитооптических дисках;
- накопители на оптических дисках с однократной записью и многократным чтением CD-R;
- накопители на оптических дисках с многократной записью и многократным чтением CD-RW;
- накопители на оптических компакт-дисках CD-ROM (Compact Disk ROM).

Cd, floppy и оптические диски – это диски, на рабочей поверхности которых информация записывается посредством прожига лазерным лучом. Такие диски, покрыты тонким напылением алюминия, имеет одну дорожку в форме спирали. Участки рабочей поверхности, поглощающие луч, воспринимаются при чтении как 0, отражающие луч – как 1.

Магнитные диски используют рабочую поверхность в виде магнитного материала. Весь магнитный диск делится на сектора и дорожки.

В последнее время появились сменные винчестеры, которые чаще всего используются в портативных компьютерах.

Логическая структура дисков. Все винчестеры и флоппи-диски, поддерживаемые MS-DOS, за рядом исключений, имеют сходный логический формат. Под *логическим форматом* понимается то, что на диске резервируются определенные области для хранения служебной информации, необходимой операционной системе для работы с этим устройством. Процесс создания и заполнения таких областей носит название *логического форматирования*. Для создания логической структуры диска используются специальные программы, входящие обычно в состав операционной системы или существующие как независимые утилиты. Заметим, что содержимое создаваемых областей может полностью или частично заполняться и изменяться не только во время форматирования, но и в процессе последующей работы с данным диском.

Практически каждый диск содержит следующие области: загрузочная запись, или сектор BR (Boot Record), две (одну) таблицы размещения файлов (FAT – File Allocation Table), корневой каталог RD (Root Directory) и область данных DA (Data Area).

Загрузочный сектор BR каждого диска занимает по определению только один сектор.

Таблица размещения файлов располагается непосредственно после загрузочной записи и имеет переменный размер (разумеется, в секторах). FAT используется для хранения сведений о размещении файлов на диске. Заметим, что эта таблица состоит из элементов (12-, 16- или 32-битных), каждый из которых соответствует определенному участку дискового пространства и присвоенным кодом характеризует его состояние: занят, свободен или имеет

дефект. В самом начале каждой таблицы FAT (первый элемент) хранится так называемый **дескриптор** (media descriptor), определяющий тип носителя (например, для жесткого диска – F8h). Отметим, что минимальным элементом, которым MS-DOS оперирует при работе с дисками, является не сектор, а **кластер**. Кластеры состоят из нескольких секторов (2, 4 и т.д.). Для дисков с магнитным носителем обычно используются две копии FAT, которые следуют одна за другой. Содержимое их полностью дублируется.

Корневой каталог диска всегда занимает строго фиксированное место – сразу за последней таблицей FAT. Он состоит из ограниченного числа записей, каждая из которых содержит информацию о файле или другом каталоге (подкаталоге), а также метке диска. Все остальное место на диске занимает область данных, содержащая файлы данных или подкаталогов.

Флоппи-диски. До настоящего времени приводами для флоппи-дисков оснащаются большинство IBM PC-совместимых компьютеров. Они используются как для архивирования и хранения небольших объемов информации, так и для ее переноса с одного компьютера на другой.

История гибкого магнитного (флоппи) диска началась с того момента, когда магнитный слой нанесли на основу, подобную той, что используется в магнитной ленте. Чтобы не поцарапать и не испачкать поверхность носителя, диск поместили в достаточно жесткий пластиковый конверт, внутри которого он мог свободно вращаться. Первые флоппи-диски имели диаметр 8 дюймов (около 200 мм) и использовались на больших и мини-компьютерах.

Заметим, что уже на первых IBM PC использовались приводы для дисков диаметром 5,25 дюйма (133 мм), которые впервые появились в 1976 году. Первоначально на одном таком диске можно было записать всего 160 Кбайт информации, причем магнитный слой был нанесен только с одной стороны основы носителя. После того как магнитный слой стали наносить на пластиковую основу с обеих сторон, емкость носителя удвоилась. Соответственно привод стал использовать уже две головки. По мере развития технологии стала увеличиваться плотность записи, появились 5-дюймовые дискеты емкостью 360 Кбайт, а затем и 1,2 Мбайта.

В последующем дискеты стали выпускаться с диаметром 3,5 дюйма (89 мм). Их емкость сначала составляла 720 Кбайт, но вскоре достигла величины 1,44 Мбайт. При такой плотности записи защита магнитного слоя становится особенно актуальной, поэтому сам магнитный диск был спрятан в прочный пластмассовый корпус, а зона контакта головок с его поверхностью закрыта от случайных прикосновений специальной шторкой, которая отодвигается только внутри накопителя.

Первый 3,5-дюймовый привод и соответствующий микрофлоппи-диск (micro floppy disk) были разработаны в 1980 году фирмой Sony. Несколько позже эта система была принята в качестве стандарта такими организациями, как ISO и ANSI.

В последнее время все более широкое применение находят **сменные жесткие диски**. Они обычно используются для двух целей:

- периодическое сохранение данных на внешнем носителе;

- обмен информацией.

Сменные жесткие диски являются надежным, быстрым и комфортабельным средством обмена информацией.

У сменного винчестера переносным является не только носитель информации, но и фактически весь дисковод, который вынимается из своих направляющих в корпусе РС. Для извлечения дисковода на передней панели имеется специальная ручка. С обратной его стороны находится адаптер, который обычно обеспечивает силовое питание и связь для приема-передачи данных.

Этот тип сменного диска не рекомендуется для частого обмена информацией по двум причинам:

1. Данные жесткие диски достаточно надежны, но только если они прочно установлены в корпусе. Они совсем не готовы к восприятию внешних весьма значительных воздействий, зачастую возникающих при их транспортировке.

2. Второе соображение является результатом практической работы. Если вы вынимаете один винчестер и вставляете другой, то, естественно, каждый раз должны записать соответствующие параметры в CMOS Setup (в случае, если речь не идет о дисках с идентичными параметрами).

Отсюда можно сделать вывод: сменные жесткие диски, главным образом, следует использовать только для целей архивации данных.

Магнитооптические накопители. Магнитооптические накопители уже получили достаточно широкое распространение, однако не настолько широкое, как хотелось бы, из-за их соотношения цена/производительность.

Floptical – составное слово, образованное от слов Floppy-диск и Optical-диск. Принцип работы этого привода ясен из названия. Floptical имеет размер 3,5" и может быть прочитан или записан на внешнем или внутреннем дисководе. Емкость таких накопителей достигает нескольких десятков мегабайт. При оптическом чтении дорожек запись осуществляется обычными магнитными средствами, как у дискет.

МО-привод (Magneto-Optical– магнитооптический) представляет собой накопитель, в основу которого положен магнитный носитель с оптическим (лазерным) управлением.

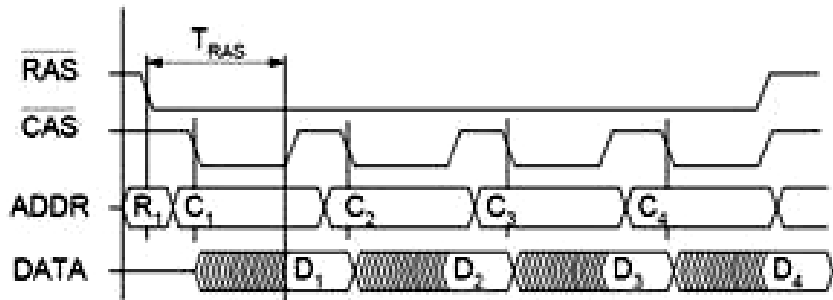
По принципу устройства МО-носитель подобен CD-ROM. Но между слоем носителя и рефлектора нанесено дополнительное напыление, которое реагирует как на оптическое, так и на магнитное воздействие. В качестве головки записи/чтения служит лазер, который нагревает отдельные участки поверхности до температуры около 150°C. Благодаря этому, элементы промежуточного слоя взаимодействуют друг с другом и после охлаждения намагничиваются. Этот процесс можно повторять любое количество раз, потому что поверхность и слой носителя защищены. Второй лазер предназначен для чтения информации.

Емкость МО-привода подобна CD-ROM. В зависимости от формата она может быть 128 Мбайт (3,5") или более 650 Мбайт (5,25"). Время доступа равно 70 мс. Средняя скорость передачи данных может достигать 700 Кбайт/с.

Основным недостатком МО-накопителей является их цена. Она все еще достаточно высока.

3 Способы повышения емкости памяти.

EDO (Extended Data Output – растянутый выход данных). В микросхеме EDO-памяти на выходе установлен буфер-защелка, фиксирующий данные после их извлечения из матрицы памяти при подъеме сигнала **CAS** и удерживающий их на выходе до следующего его спада. Это позволило сократить длительность сигнала **CAS** и соответственно цикл памяти.



Способы повышения быстродействия ИС памяти DDR SDRAM, обеспечивающая двойную скорость передачи данных (DDR – Double или Dual Data Rate), в которой за один такт осуществляются две передачи данных – по переднему и заднему фронтам каждого синхроимпульса.

Контрольные вопросы:

1. Что называют запоминающим устройством (ЗУ)?
2. Приведите классификацию ЗУ по различным признакам.
3. Назовите важнейшие параметры ЗУ.
4. Назовите наиболее распространенные типы накопителей.
5. Способы организации памяти.
6. Поясните логическую структуру диска.
7. Как достигается увеличение разрядности и количества слов в памяти?

Тема 5. Оперативная память (RAM)

1. Оперативная память (RAM).
2. Запоминающие устройства статического (SRAM) и динамического (DRAM) типов.
3. Контроллер динамической памяти.
4. Кэш-память. Кэширование оперативной памяти.

1 Оперативная память (RAM)

Оперативная память (ОЗУ) или RAM (Random Access Memory) относится к одному из важнейших компонентов ЭВМ. От ее параметров в значительной степени зависят функциональные возможности всей системы. Она позволяет не только считывать хранящиеся в ней данные, но и записывать новые. В микросхемах оперативной памяти хранится информация, которая меняется в процессе работы ПК. RAM отличается высокой скоростью записи и чтения. При выключении ПК данные, хранящиеся в RAM, теряются. Объем оперативной памяти зависит от количества и типа используемых микросхем.

Характеристики оперативной памяти и особенности ее устройства являются важнейшим фактором, от которого зависит быстродействие компьютера. Даже при наличии «быстрого» процессора скорость выборки данных из памяти может оказаться невысокой, и именно эта невысокая скорость работы ОЗУ будет определять быстродействие ЭВМ. Время цикла работы с памятью t_m особенно заметно больше чем время цикла работы центрального процессора t_c . При обращении к ОЗУ, время обращения $T=t_m+t_c$, в период этого времени ни одно устройство не может работать. Таким образом, возникает проблема доступа к ОП.

Оперативная память выполнена обычно на микросхемах динамического типа с произвольной выборкой DRAM (Dynamic Random Access Memory). Каждый бит такой памяти представляется в виде наличия (или отсутствия) заряда на конденсаторе, образованном в структуре полупроводникового кристалла. В современных ПК DRAM используется обычно в качестве оперативной памяти общего назначения, а также как память видеоадаптера. Из применяемых типов динамической памяти наиболее известны: DRAM, FPM DRAM, EDO DRAM, BEDO DRAM, ED RAM, CD RAM, Synchronous DRAM, DDR SDRAM и SL RAM; видеопамять: MDRAM, VRAM, WRAM, SGRAM, RDRAM и др.

Другой, более дорогой тип памяти – статический. SRAM (Static RAM) в качестве элементарной ячейки использует так называемый статический триггер, схема которого состоит из нескольких транзисторов. Статический тип памяти обладает более высоким быстродействием. SRAM обычно применяется в качестве КЭШ-памяти второго уровня для кэширования основного объема ОЗУ. Такая память выполняется на основе ТТП, КМОП или БиКМОП – микросхем. По способу доступа к данным может быть как асинхронной, так и синхронной.

Асинхронным называется доступ к данным, который можно осуществлять в произвольный момент времени. SRAM данного типа применялась на материнских платах третьего ÷ пятого поколения процессоров. Время доступа к ячейкам такой памяти составляло от 15нс (33МГц) до 8нс (66МГц).

Для описания характеристик быстродействия ОЗУ применяются так называемые циклы чтения/записи. При обращении к памяти на считывание или запись первого машинного слова расходуется больше тактов, чем на обращение к трем последующим словам.

Для асинхронной SRAM чтение одного слова выполняется за три такта, запись за четыре такта. Чтение нескольких слов определяется последовательностью: 3 – 2 – 2 – 2, а запись: 4 – 3 – 3 – 3.

Синхронная SRAM обеспечивает доступ к данным не в произвольные моменты времени, а синхронно с тактовыми импульсами. В промежутках между ними память может готовить для доступа следующую порцию данных. В большинстве материнских плат пятого поколения используется разновидность синхронной памяти – синхронно-конвейерная SRAM (Pipelined- Burst SRAM). Для которой типичное время одной операции чтения/записи составляет три такта, а групповая операция занимает 3 – 1 – 1 – 1 такта при первом обращении и 1 – 1 – 1 – 1 при последующих обращениях, что обеспечивает ускорение доступа более чем на 30%.

Логическое распределение оперативной памяти определяется применяемой операционной системой и особенностями аппаратной реализации IBM-совместимых РС. Знание строения ОЗУ необходимо для программной оптимизации системы.

Можно выделить пять важнейших логических областей оперативной памяти:

- Стандартная оперативная память (Conventional Memory);
- EMS-память (Expanded Memory Specification) – дополнительная память;
- UMA (Upper Memory Area) – верхняя память;
- HMA (High Memory Area);
- XMS (Extended Memory Specification) – расширенная память

С точки зрения аппаратной спецификации, стандартная память является наиважнейшей. Как правило, в ней располагается большая часть прикладных программ и данных.

Стандартная память начинается от адреса 0000:0000 и продолжается до адреса A000:0000, занимая 640Кбайт. Этот верхний предел был установлен процессорами 8088 и 8086, которые имели 20-разрядную адресную шину и потому могли адресовать только 1 Мбайт памяти ($2^{20}=1\ 048\ 576$ байт). В пределах этой памяти выше уровня 640 Кбайт фирмой IBM были зарезервированы 384 Кбайт для выполнения внутренних функций.

В стандартной памяти размещены таблица векторов прерываний, область данных BIOS и операционная система.

2 Запоминающие устройства статического (SRAM) и динамического (DRAM) типов

В статических ЗУ (*Static Random Access Memory* – SRAM) в качестве элемента памяти используется триггер, что, конечно, сложнее, чем конденсатор с транзисторным ключом динамического ЗУ. Поэтому статические ЗУ обладают меньшей плотностью хранения информации: емкость типовых микросхем статических ЗУ начала 2000-х годов не превосходила 16 Мбит.

Однако триггер со времен первых компьютеров был и остается самым быстродействующим элементом памяти. Поэтому статическая память позволяет достичь наибольшего быстродействия, обеспечивая время доступа в единицы и даже десятые доли наносекунд, что и обуславливает ее использование в ЭВМ, главным образом, в высших ступенях памяти – кэш-памяти всех уровней.

Главными недостатками статической памяти являются ее относительно высокие стоимость и энергопотребление.

Конечно, в зависимости от используемой технологии, память будет обладать различным сочетанием параметров быстродействия и потребляемой мощности. Например, статическая память, изготовленная по КМОП-технологии (CMOS память), имеет низкую скорость доступа, со временем порядка 100 нс, но зато отличается очень малым энергопотреблением. В ПЭВМ такую память применяют для хранения конфигурационной информации компьютера при выключенном напряжении сети (в этой же микросхеме размещают и часы, отсчитывающие реальное время). Питание такой памяти осуществляется от небольшой батарейки, которая может служить несколько лет.

Основными разновидностями статической памяти (SRAM) с точки зрения организации ее функционирования являются асинхронная (*Asynchronous*), синхронная пакетная (*Synchronous Burst*) и синхронная конвейерно-пакетная (*Pipeline Burst*) память.

Первой появилась асинхронная память, Интерфейс этой памяти включает шины данных, адреса и управления. В состав сигналов последней входят:

CS# (Chip Select) – сигнал выбора микросхемы;

WE# (Write Enable) – сигнал разрешения записи;

OE# (Output Enable) – сигнал включения выходов для выдачи данных.

Все сигналы управления инверсные, т.е. их активный (вызывающий соответствующее действие) уровень низкий. При единичном значении сигнала *OE#* выход микросхемы переходит в состояние высокого выходного сопротивления.

Временные диаграммы циклов чтения и записи приведены на рис. 1 а) и не требуют особых пояснений. Цикл записи может быть организован и несколько иначе, чем показано на рис. 1 б), в случае удержания во время цикла высокого уровня сигнала *OE#*.

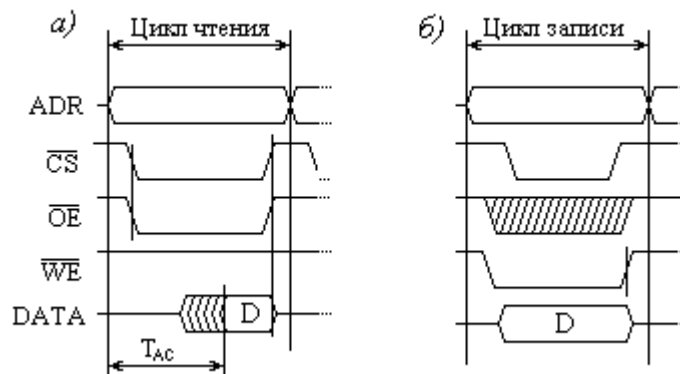


Рисунок 1 – Временная диаграмма простых циклов чтения а) чтения и б) записи

Время доступа t_{AC} у типовых микросхем составляет порядка 10 нс. Поэтому реально такие микросхемы могут работать на частотах, близких к частоте системной шины, только если эти частоты не превышают 66 МГц.

Несколько позже появилась синхронная пакетная статическая память (SBSRAM), ориентированная на выполнение пакетного обмена информацией, который характерен для кэш-памяти. Эта память включает в себя внутренний счетчик адреса, предназначенный для перебора адресов пакета, и использует сигналы синхронизации CLK , как и синхронная DRAM память.

Для организации пакетного обмена, помимо имеющихся у асинхронной памяти управляющих сигналов $CS\#$, $OE\#$ и $WE\#$, в синхронную память также введены сигналы $ADSP\#$ (*Address Status of Processor*) и $CADS\#$ (*Cache Address Strobe*), сопровождающие передачу адреса нового пакета, а также сигнал $ADV\#$ (*Advance*) продвижения на следующий адрес пакета. Пакетный цикл всегда предусматривает передачу четырех элементов, так как внутренний счетчик имеет всего 2 бита, причем перебор адресов в пределах пакета может быть последовательным или с расслоением (чередованием) по банкам (при использовании процессоров семейства x86).

Временные диаграммы пакетных циклов чтения и записи приведены на рис. 2. Обращения к синхронной памяти могут быть и одиночными. В этом случае низкому уровню сигнала $ADSP\#$, указывающему на передачу адреса, соответствует высокий уровень сигнала $CADS\#$, а не низкий, как при пакетном цикле. Параметр T_{OQ} характеризует время задержки данных относительно синхронизирующего сигнала.

Следующим шагом в развитии статической памяти явилась конвейерно-пакетная память PBSRAM, обеспечивающая более высокое быстродействие, чем SBSRAM. В нее были введены дополнительные внутренние буферные регистры данных (здесь можно провести аналогию с EDO DRAM памятью) адреса, а в ряде модификаций предусмотрена возможность передачи данных на двойной скорости по переднему и заднему фронтам синхросигнала и используются сдвоенные внутренние тракты записи и чтения. Это позволило

получить время обращения порядка 2-3 нс и обеспечить передачу данных пакета без задержек на частотах шины более 400 МГц.

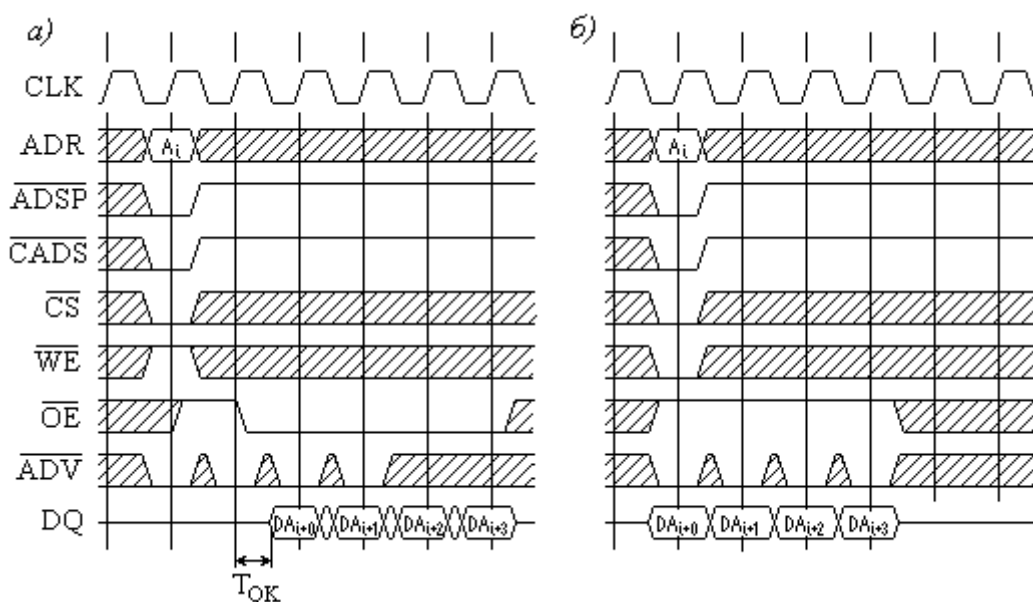


Рисунок 2 – Пакетные циклы чтения *а)* записи *б)* синхронной пакетной памяти (SBSRAM)/

Внутренняя логика позволяет переключаться с циклов чтения на циклы записи и наоборот без дополнительных задержек, кроме того, анализируется совпадение адресов записи и чтения для исключения избыточных операций.

Структурная схема такой памяти приведена на рис. 2.3, где ФАП – блок формирования адресов пакета, МП – мультиплексоры, переключающие внутренние тракты чтения и записи в соответствии со значением младшего разряда адреса A_0 .

Временная диаграмма, приведенная на рис. 2.4, показывает запись и чтение как на одиночной (SDR – *Single Data Rate*), так и на двойной скорости (DDR – *Double Data Rate*) передачи. Сигналы CQ и $CQ\#$ – дифференциальные выходные сигналы синхронизации, близкие по времени к моменту появления данных на шине при чтении и используемые для синхронизации принимающих устройств. Сигналы SA и B – адресные и управляющие сигналы соответственно, причем последние используются для задания типа цикла. В режиме чтения с двойной скоростью передачи формируются два набора данных, начиная со второго переднего и заднего фронтов синхросигнала, если по его первому переднему фронту передается начальный адрес пакета. Первый набор данных (DOUT-A) формируется для заданного адреса, а второй (DOUT-A') – для следующего адреса пакета, в соответствии с определенным для пакета порядком.

Аналогичным образом запись с двойной скоростью передачи требует установки данных для записи, начиная со второго переднего и заднего фронтов сигнала синхронизации.

В режиме чтения с одиночной скоростью формируется только один набор данных по переднему фронту синхросигнала, начиная со второго сигнала, а при

записи с такой же скоростью данные должны выставляться на шину, начиная с переднего фронта второго синхросигнала после передачи адреса.

Переключение из режима двойной скорости в режим одиночной (и наоборот) производится при подаче соответствующего управляющего сигнала.

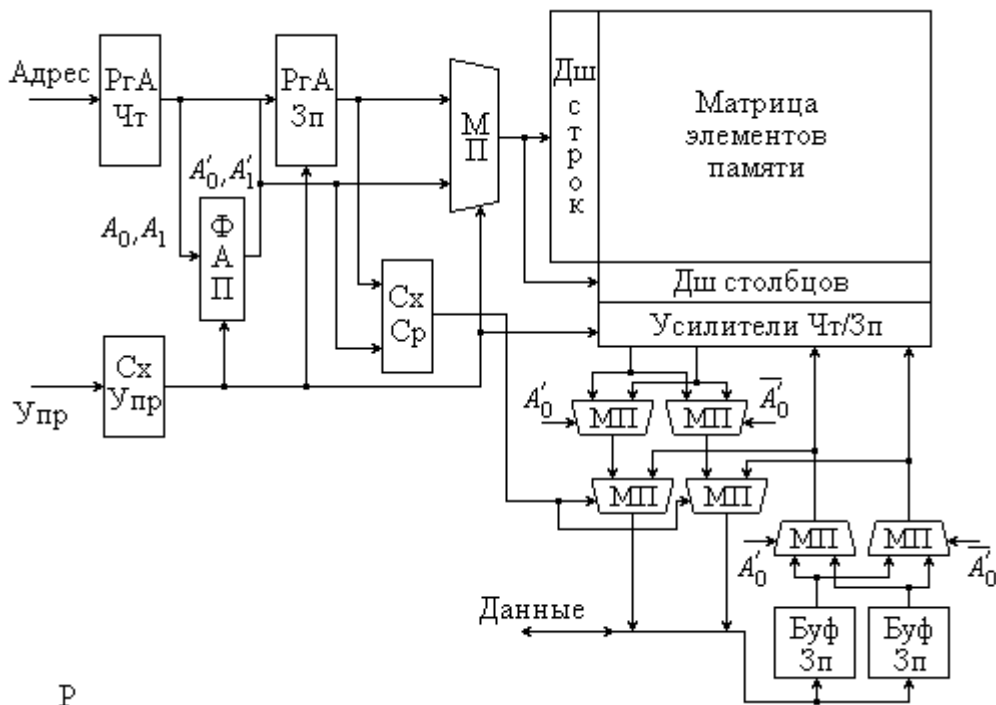


Рисунок 3 - Структурная схема синхронной статистической пакетно-конвейерной (PBSRAM) памяти

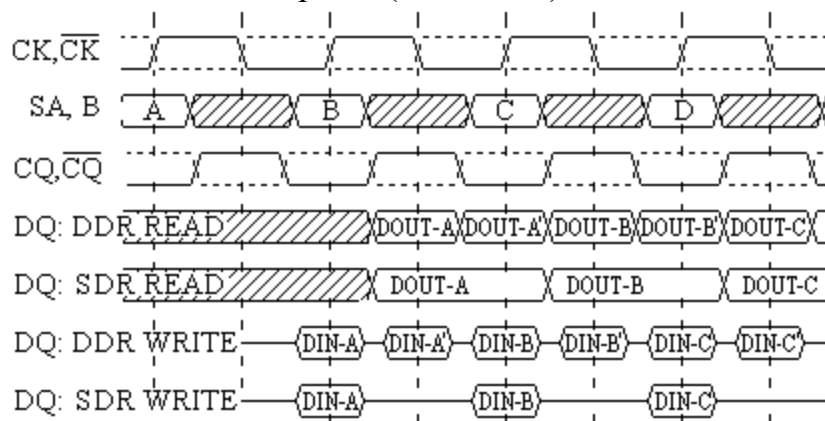


Рисунок 4 - Общий вид диаграмм чтения и записи с одиночной и двойной скоростью передачи в синхронной конвейерно-пакетной статистической памяти (PBSRAM)

Динамическая память - DRAM (Dynamic RAM) - получила свое название от принципа действия элемента памяти, хранящего один бит информации. Запоминающим элементом служит конденсатор, являющийся емкостью перехода исток - сток полевого транзистора. При записи логической единицы в элемент памяти конденсатор заряжается, при записи нуля - разряжается. При

чтении информации конденсатор тоже разряжается и, если заряд был не нулевым, усилитель считывания устанавливает на выходе единичное значение потенциала, и конденсатор специальной схемой перезаписи снова заряжается до потенциала логической единицы. При отсутствии обращения к элементу динамической памяти со временем, за счет токов утечки, конденсатор разряжается и информация теряется, поэтому такая память требует постоянного периодического перезаряда конденсаторов методом перезаписи каждого элемента (регенерации). Следовательно, память может работать только в динамическом режиме, с постоянным перезарядом конденсаторов ячеек памяти. Этим она принципиально отличается от статической памяти, реализуемой на триггерных ячейках и хранящей информацию без обращений к ней статично, сколь угодно долго при включенном питании. Благодаря относительной простоте ячейки динамической памяти, на одном кристалле удастся размещать миллионы ячеек и получать сравнительно недорогую полупроводниковую память, пускай и недостаточно высокого быстродействия, но с умеренным энергопотреблением. Используется динамическая память в качестве оперативной памяти компьютера и оперативной памяти видеоадаптера.

Микросхема динамической памяти состоит из миллионов элементов, каждый из которых хранит всего один бит информации. На физическом уровне элементы памяти объединяются в прямоугольную матрицу, горизонтальные линейки элементов называются строками (ROW), а вертикальные - столбцами (Column).

Элементы всей строки матрицы могут считываться в современных микросхемах динамической памяти в буферный регистр - этот набор ячеек принято называть страницей (Page). Из-за высокой интеграции микросхем памяти становится невозможным обратиться к ячейке памяти, установив на шине адреса адрес ячейки памяти полностью. Слишком много разрядов должна иметь эта шина. Поэтому шину адреса делают в два раза уже и устанавливают адрес дважды. Сначала адрес строки, затем адрес столбца. Чтобы различать, установка какой части адреса происходит, вводятся дополнительные сигналы управления RAS и CAS. Адрес строки передается по шине адреса и принимается в регистр адреса строки микросхемы памяти по спаду импульса RAS (Row Access Strobe). Адрес колонки передается по этой же шине и принимается в регистр адреса колонки микросхемы памяти по спаду импульса CAS (Column Access Strobe). На пересечении линии строки и колонки находится элемент памяти.

Поскольку обращение (запись или чтение) к различным ячейкам памяти обычно происходит в случайном порядке, то для поддержания сохранности данных производится регулярная регенерация (Memory Refresh - "освежение" памяти) - регулярный циклический перебор ячеек памяти с холостыми циклами перезаписи. Регенерация в микросхеме происходит одновременно по всей строке матрицы при обращении к любой из ее ячеек.

3 Контроллер динамической памяти

Главные особенности микросхем динамической памяти заключаются в следующем:

- отсутствуют источники для питания элементов запоминающих ячеек;
- необходимы логические схемы, обеспечивающие регенерацию информации;
- максимально проста схема накопителя, обеспечивающая минимум занимаемой площади;
- мала потребляемая мощность, поскольку динамический запоминающий элемент не потребляет тока в отрезки времени, когда к нему не происходит обращение.

Для формирования внутренних сигналов, управляющих включением и выключением в определенной последовательности различных узлов микросхемы, в структуре микросхемы предусмотрена схема управления. Входные, выходные и управляющие сигналы микросхемы памяти: RAS (Row Address Strobe) - строб адреса строки; CAS (Column Address Strobe) - строб адреса столбца; W/R (запись/чтение); вход данных - DI; выход данных - DO. Микросхема имеет усилители считывания и регенерации в каждом столбце матрицы, регистр адреса, дешифраторы адреса строк и адреса столбцов, устройство управления и устройство ввода и вывода данных. Структура микросхемы приведена на рисунке 6.

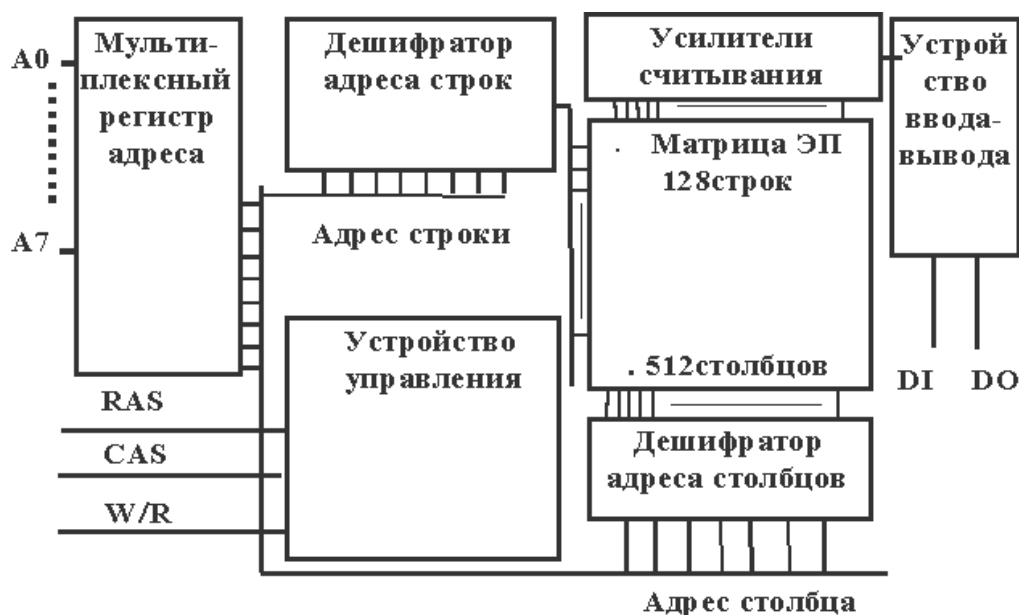


Рисунок 6 - Структура микросхемы динамической памяти

На линиях RAS (row address strobe - строб адреса строки) и CAS (column address strobe - строб адреса столбца) в спокойном состоянии поддерживается высокий уровень сигнала, что означает, что никакой информации на адресных линиях нет и никаких действий, связанных с чтением или записью данных. В

случае строчной регенерации, регенерация информации осуществляется путем обращения к каждой строке динамической памяти перебором строчных адресов в любом режиме работы.

В современных микросхемах памяти регенерацию производит устройство регенерации, встроенное в микросхему памяти. Нарушение режима регенерации может привести к нарушению логических состояний микросхем памяти. Поэтому в схемах регенерации применяются защитные схемы, поддерживающие истинность логических состояний элементов матрицы микросхемы памяти, совместно с генератором циклов регенерации. Микросхема подключается к шине адреса и данных. Линии адреса служат для выбора адреса ячейки памяти, а линии данных - для чтения или записи данных в память.

Низкий уровень сигнала W/R готовит микросхему к считыванию состояния шины данных и записи полученной информации в соответствующую ячейку, а высокий, наоборот, заставляет считать содержимое ячейки и установить его на линиях шины данных. Использование одной и той же шины данных для вывода данных и для чтения данных позволяет в два раза уменьшить количество выводов шины данных. Это, как и уменьшение ширины шины адреса позволяет уменьшить размеры микросхемы, а значит, получить более высокие рабочие частоты.

Рассмотрим, какие факты мешают повышению рабочей частоты микросхемы памяти:

1. Скорости распространения электрических сигналов ограничены, поэтому длины проводников, подведенных к различным ножкам микросхемы, не должны сильно отличаться друг от друга, иначе сигнал от одного вывода будет опережать сигнал от другого.

2. Длины проводников не должны быть очень велики, задержка распространения сигнала не позволит повысить быстродействие.

3. Любой проводник действует как приемная и как передающая антенна, создавая помехи. Уровень помех резко усиливается с ростом тактовой частоты.

4. Любой проводник обладает электрической емкостью. Чем больше емкость, тем меньше скорость передачи данных.

Совмещение выводов шины для вывода и ввода данных увеличивает скорость обмена с памятью, но не позволяет осуществлять чтение и запись одновременно.

Интересно, что в первых компьютерах специальная схема приостанавливала (прерывала) работу процессора на время регенерации, для управления процессом регенерации.

Микросхемы памяти могут иметь и другую организацию. Например, распространенные (2006г.) микросхемы (чипы) синхронной динамической памяти DDR SDRAM фирмы Samsung: K 4 H 5 6 0 8 3 8 E — T C B 3 (рис. 7) имеют емкость 36 мегабайт (256 мегабит). Адресуемая ячейка памяти чипа 8 битная.



Рисунок 7 - Микросхема памяти DDR SDRAM фирмы Samsung

Для того, чтобы исключить ожидания при регенерации считанных данных, современные микросхемы памяти имеют несколько матриц. Последовательные обращения к данным при чтении будут происходить без задержек, если последовательные ячейки памяти расположены на разных матрицах. Таким образом, можно избежать задержек на регенерацию при чтении последовательных данных. Рассматриваемая нами микросхема имеет 4 матрицы (допускает при построении модуля памяти использование четырех банков памяти).

Контроллер динамической оперативной памяти осуществляет передачу команд центрального процессора к оперативной памяти, преобразование этих команд в формат, необходимый для обращения к конкретной группе ячеек оперативной памяти. Именно контроллер оперативной памяти является ответственным элементом за все операции, которые производит процессор с оперативной памятью. А операции эти такие: запись данных в оперативную память, чтение данных из оперативной памяти, модификация данных в оперативной памяти.

Контроллер динамической оперативной памяти состоит из таких элементов:

- устройство управления;
- устройство записи;
- устройство считывания;
- дешифратор строк;
- дешифратор столбцов.

4 Кэш-память. Кэширование оперативной памяти

Мы уже не раз отмечали, что оперативная память вычислительных систем реализуется на относительно медленных микросхемах динамической памяти. Если процессор работает непосредственно с памятью такого типа, требуется вводить такты ожидания для процессора. В этом случае процессор частично простаивает. Статическая память, построенная на триггерах, имеет то же быстродействие, что и процессоры и использование такой памяти позволило бы работать процессору без тактов ожидания, но параметр цена/производительность диктует поиски другого выхода из этой ситуации. Разумным компромиссом для построения экономичных и производительных систем явился иерархический способ построения памяти.

Обмен процессора с оперативной памятью производится при помощи промежуточной относительно небольшой кэш-памяти на быстродействующих микросхемах SRAM.

В переводе слово кэш (cache) означает склад или тайник. Для выполняемой программы он не представляет собой дополнительной адресуемой памяти. Он выполняет роль быстродействующего хранилища копий блоков информации основной памяти, к которым в ближайшее время ожидается обращение. Объем кэш во много раз меньше объема оперативной памяти. Контроллер кэш памяти ведет каталог и при каждом обращении к кэш памяти по каталогу проверяет, есть ли действительная копия затребованных данных в кэше? Если она там есть, то это случай кэш попадания (cache hit), и обращение за данными происходит только к кэш-памяти. Если действительной копии там нет, то это случай кэш-промаха (cache miss), и данные берутся из основной памяти.

Отношение числа попаданий к общему числу обращений к памяти называется коэффициентом эффективности поиска. Если этот коэффициент достаточно высок, большая часть времени затрачивается на обращение к кэш-памяти, что эквивалентно использованию быстродействующей основной памяти при выполнении программы. Обычно коэффициент эффективности поиска превышает 95%.

Обращение к основной памяти может начинаться одновременно с поиском данных в каталоге кэш, и в случае попадания - прерывается (архитектура Look Aside). При параллельном поиске из - за параллельного обращении к оперативной памяти возрастают затраты энергии.

В другом варианте обращение к внешней памяти начинается только после фиксации случая промаха (архитектура Look Through), на этом теряется, по крайней мере, один такт процессора, зато значительно экономится энергия.

В современных процессорах кэш обычно строится по двухуровневой схеме. Первичный кэш (L1 Cache) имеет сравнительно небольшой объем и для повышения производительности обычно используется отдельный кэш для данных и команд (так называемая Гарвардская архитектура - противоположность Принстонской, использующей общий кэш для команд и данных). Вторичный кэш (L2 Cache) тоже включается в микросхему микропроцессора, но не разделяется на кэш команд и данных. Работает встроенный кэш на внутренней тактовой частоте процессора.

Алгоритм сквозной записи (WT) предусматривает выполнение каждой операции записи (даже однобайтной) одновременно и в строку кэш памяти, и в основную память. Алгоритм достаточно прост в реализации и легко обеспечивает целостность данных за счет постоянного совпадения копий данных в кэш и в основной памяти. Но эта простота оплачивается низкой эффективностью записи.

Алгоритм обратной записи (WB) позволяет уменьшить количество операций записи в оперативную память. Если блок памяти, в который должна производиться запись, отображен и в кэш памяти, то физическая запись сначала будет произведена в эту действительную строку кэш памяти, и будет отмечена как грязная (dirty), или модифицированная, то есть требующая записи в

оперативную память. Только после записи в оперативную память строка станет чистой (clean). Данный алгоритм сложнее в реализации, но существенно эффективнее, чем алгоритм сквозной записи (WT).

В зависимости от способа определения взаимного соответствия строки кэш памяти и области основной памяти различают три архитектуры кэш-памяти: кэш прямого отображения (direct-mapped cache), полностью ассоциативный кэш (fully associative cache) и их комбинация - частично или наборно-ассоциативный кэш (set-associative cache).

Организация кэш прямого отображения

Кэш прямого отображения (direct mapped) наиболее дешевый и простой по организации. Область оперативной памяти разбивается на блоки. На такие же блоки разбивается кэш память. Количество блоков в кэш значительно меньше, чем количество блоков в ОЗУ. Каждый блок основной памяти имеет только одно фиксированное место, на котором он может появиться в кэш-памяти.

При записи данных в кэш каждый блок оперативной памяти может претендовать только на определенный блок кэш памяти. Например, на месте первого блока кэш памяти, состоящей из четырех блоков прямого отображения (рис. 3.1), может размещаться первый блок ОЗУ, пятый, девятый и т.д. На место второго блока в кэш претендуют блоки 2-ой, 6-ой, 10-ый и т.д. Как видим, на одну и ту же область кэш претендует несколько блоков ОЗУ, что может привести к конфликтам.

Кроме того, даже если конфликты разрешаются схемой управления, то при последовательном обращении к блокам, претендующим на одно и то же место в кэш, происходит постоянная замена блоков и кэш память теряет смысл, т.к. не увеличивает скорость обмена данными между процессором и оперативной памятью.

Полностью ассоциативный кэш

Полностью ассоциативный кэш реализуется аппаратно. Поиск информации, как и в ассоциативной памяти производится по ключевому слову. Особенность такой организации кэш в том, что любой блок оперативной памяти может занимать любое место в кэш памяти. Такой способ адресации не требует внешнего описания памяти. Информация о том, какой именно блок занимает данную строку, и каково состояние строки (действительная или пустая) называется тегом (tag) и хранится в связанной с данной строкой ячейке специальной памяти тегов (tag RAM). Поиск данных производится параллельно во всех блоках кэш. При операции записи или считывания контроллер кэш проверяет нет ли записи в кэш и фиксирует частоту обращения к каждому блоку кэш. Такая реализация требует больших аппаратных затрат. Для персональных компьютеров этот способ реализации кэш слишком дорог.

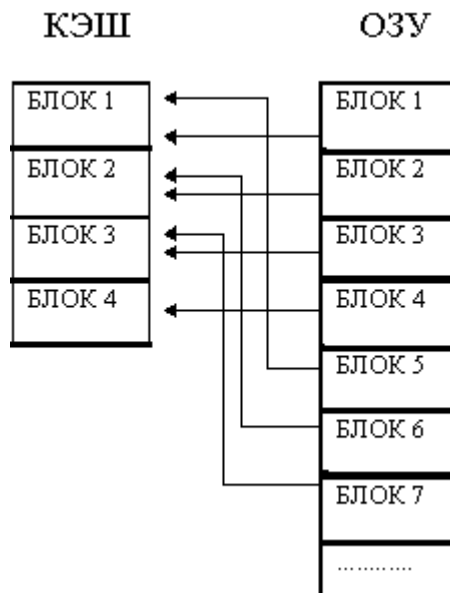


Рисунок 8 - Кэш прямого отображения

Адресно-ассоциативный кэш

Для поиска данных используется адрес колонки. Кэш память и оперативная память разбиваются на строки (страницы) и колонки. Количество колонок кэш памяти и оперативной памяти строго совпадает. Поиск информации в Кэш памяти происходит по строкам путем сравнения адресов строк.

Обмен данными производится блоками, например по 64 байта. Схема сравнения выявляет наличие блока в кэш - памяти по ассоциативному признаку. Если блок присутствует, то он читается из кэш, если нет, то данные читаются из ОЗУ, и одновременно отсутствующий блок записывается в кэш.

Для ускорения работы с кэш вводят многовходовые или ассоциативные по множеству кэш, в которых кэш память разбивается на N частей. При обращении к N-входовому кэш информация ищется одновременно в N ячейках кэш.

Ассоциативный по множеству кэш

Ассоциативный по множеству кэш представляет собой N-ое количество малых кэш прямого отображения, в которых жестко распределено закрепление отображаемых блоков оперативной памяти. Например, четырехвходовой частично ассоциативный кэш содержит четыре блока данных в которых одновременно производится поиск по ассоциативному признаку. Замена блока производится только в одном из четырех блоков кэш.

Рассмотрим пример организации 4-х входного кэш.

Кэш разбивается на 4 части. Для каждой части вводится определитель, по которому производится поиск данных (ассоциативный признак). Выделяется байт, который фиксирует обращение к блоку данных. Фиксируется активность каждого блока для выявления и удаления неактивного блока. Когда данные не обнаруживаются в кэш памяти, происходит обращение к ОЗУ, неактивный блок заменяется новым.

При обращении к 4-х входному кэш наличие информации определяется одновременно в 4-х блоках. При обращении к многовходовому кэш-адрес делится на две части. Одна часть содержит поле индекса и выбирает строку кэш, вторая содержит поле признаков, которое сравнивается с ассоциативным признаком ячейки памяти, используемым при поиске информации.

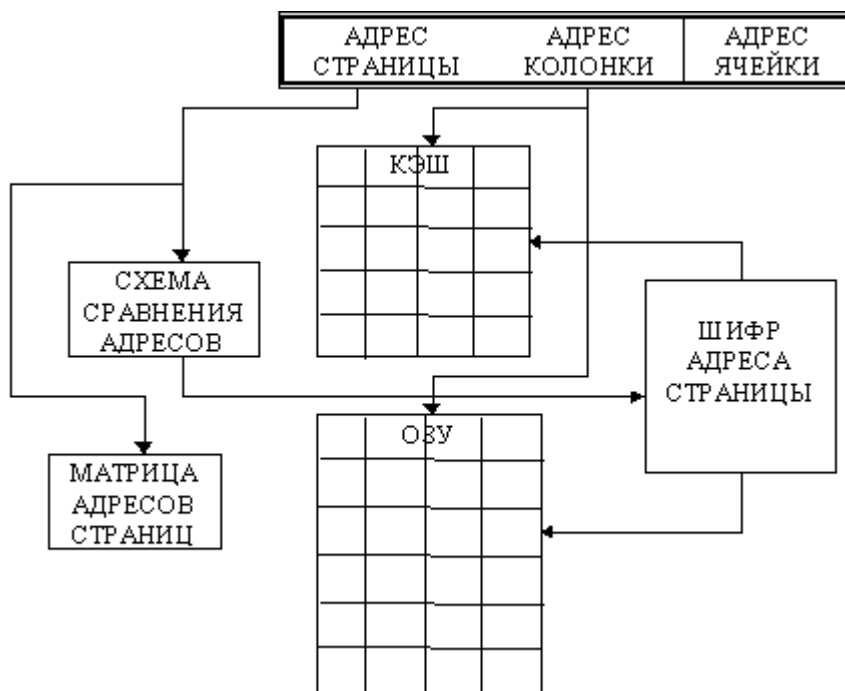


Рисунок 9 – Схема работы КЭШа

Сегодня разработчики высокопроизводительных микропроцессоров часто предпочитают этот тип кэш. Это обусловлено тем, что обычно частично-ассоциативный кэш обеспечивает более высокую вероятность нахождения в нем данных по сравнению с прямоадресуемым (прямого отображения) кэш той же емкости. Когда прямоадресуемый кэш заполняется, возникают проблемы, куда положить очередную порцию запрошенных данных. В n-канальном частично-ассоциативном кэш (n-way set-associative) имеется до n мест, куда можно положить запрошенные данные. Для более ясного понимания ситуации полезно рассмотреть гипотетический пример, предложенный на одной из телеконференций Usenet.

Пусть вероятность попадания в прямоадресуемый кэш L2 составляет 80%, а в частично-ассоциативный кэш L1 - порядка 85-90%, скажем 87%. Пусть попаданию в кэш отвечает задержка 20 нс для прямоадресуемого кэша и 24 нс - для частично-ассоциативного кэша, а непопаданию в кэш - 100 нс. Тогда средние задержки будут составлять:

$$0,80 * 20 + 0,20 * 100 = 36 \text{ нс. (для прямоадресуемого кэш)}$$

$$0,87 * 24 + 0,13 * 100 = 34 \text{ нс. (для частично-ассоциативного кэш).}$$

Для управления работой кэш памяти используется конвейер данных. Работа конвейеров кэш-памяти данных тесно координирована. Например,

команды загрузки могут выполнять проверку тегов и чтение данных в том же такте, что и преобразование адреса. Команды записи сразу же начинают проверку тегов, чтобы в случае необходимости как можно раньше инициировать заполнение требуемой строки из кэш-памяти второго уровня, но непосредственная запись данных в кэш задерживается до тех пор, пока сама команда записи не станет самой старой командой в общей очереди выполняемых команд и ей не будет позволено зафиксировать свой результат. Промах при обращении к кэш памяти данных первого уровня инициирует процесс заполнения строки из кэш памяти второго уровня. При выполнении команд загрузки одновременно с заполнением строки кэш-памяти данные могут поступать по цепям обхода в регистровый файл. При обнаружении промаха при обращении к кэш-памяти данных ее работа не блокируется, т.е. она может продолжать обслуживание следующих запросов.

Для обеспечения целостности данных в кэш-памяти большой емкости обычной практикой является использование кодов, исправляющих одиночные ошибки (ЕСС-кодов).

Любой внутренний запрос процессора на обращение к памяти направляется во внутренний кэш. Теги строк [кэш-памяти] сравниваются со старшими битами запрошенного физического адреса. Если адресуемая область представлена в строке кэш-памяти - случай попадания (Cache Hit), запрос на чтение обслуживается только кэш-памятью, не выходя на внешнюю шину. Запрос на запись модифицирует данную строку и в зависимости от политики записи на внешнюю шину выходит либо сразу (при сквозной записи), либо несколько позже при использовании алгоритма обратной записи.

В случае промаха (Cache Miss) запрос на запись направляется только на внешнюю шину, а запрос на чтение обслуживается сложнее. Если этот запрос относится к кэшируемой области памяти, выполняется цикл заполнения целой строки кэша. Данные читаются из оперативной памяти и помещаются в одну из строк набора кэша. Если затребованные данные не укладываются в одной строке, заполняется и соседняя.

Внутренний запрос на данные удовлетворяется сразу, как только затребованные данные считываются из ОЗУ - заполнение строки до конца может происходить параллельно с обработкой полученных данных. Если имеется свободная строка (с нулевым битом достоверности), заполнена будет именно она и для нее установится бит достоверности. Если свободных строк в наборе нет, будет замещена строка, к которой дольше всех не было обращений. Выбор строки для замещения выполняется на основе анализа бит *LRU* (Least Recently Used) по алгоритму "псевдо-*LRU*". Эти биты модифицируются при каждом обращении к строке данного набора (кэш-попадании или замещении). Таким образом, выделение и замещение строк выполняется только для кэш-промахов чтения, при промахах записи заполнение строк не производится. ... Кроме того, существует возможность их аннулирования - объявления недостоверными и очистки всей кэш-памяти. Управление заполнением кэш возможно и на аппаратном, и на программном уровнях.

Контрольные вопросы:

1. Назначение оперативной памяти.
2. Какие ЗУ называют статическими?
3. Что используется в качестве элемента памяти статических ЗУ?
4. Перечислите основные разновидности статической памяти (SRAM).
5. Какая технология изготовления статических ОЗУ наиболее интенсивно развивается в настоящее время?
6. Что называют динамической памятью?
7. Как работают запоминающие элементы динамической памяти?
8. На основе чего строятся микросхемы динамической памяти?
9. В чем заключаются главные особенности микросхем динамической памяти?
10. Поясните структуру микросхемы динамической памяти?
11. Какую функцию выполняет контроллер динамической оперативной памяти?
12. Из каких элементов он состоит?
13. Назначение кэш-памяти.
14. Как в современных процессорах строится кэш?
15. Как может быть построена кэш-память?

Тема 6. Масочные, программируемые и репрограммируемые постоянные запоминающие устройства (ROM, PROM, EPROM, EEPROM). Флэш-память. Память с последовательным доступом. Видеопамять. Буферная память типа FIFO. Стековая память (LIFO).

1. Масочные, программируемые и репрограммируемые постоянные запоминающие устройства (ROM, PROM, EPROM, EEPROM).
2. Флэш-память.
3. Память с последовательным доступом.
4. Видеопамять.
5. Буферная память типа FIFO.
6. Стековая память (LIFO).

1 Масочные, программируемые и репрограммируемые постоянные запоминающие устройства (ROM, PROM, EPROM, EEPROM)

Постоянная память является энергонезависимой, информация в ней сохраняется и после выключения питания компьютера.

Существует множество типов энергонезависимой памяти: ROM, PROM, EPROM, EEPROM, Flash Memory, различающихся по своим свойствам, обусловленным способом построения запоминающих ячеек, и сферам применения. Запись информации в энергонезависимую память, называемая программированием, обычно существенно сложнее и требует больших затрат времени и энергии, чем считывание. Основным режимом работы такой памяти является считывание данных, а некоторые типы после программирования допускают только считывание, что и обуславливает их общее название ROM (Read Only Memory - память только для чтения) или ПЗУ (постоянное запоминающее устройство). Самые первые постоянные запоминающие устройства выполнялись на магнитных сердечниках, где информация заносилась их прошивкой проводниками считывания. С тех пор применительно к программированию ПЗУ укоренилось понятие "прошивка".

По возможности программирования различают:

- Микросхемы, программируемые при изготовлении (масочные ПЗУ) - ROM.
- Микросхемы, программируемые однократно после изготовления перед установкой в целевое устройство (прожигаемые ПЗУ, программируемые на специальных программаторах) - *PROM* (Programmable ROM) или ППЗУ (программируемые ПЗУ).
- Микросхемы, стираемые и программируемые многократно - РПЗУ (репрограммируемые ПЗУ), *EPROM* (Erasable PROM - стираемые ПЗУ).

Запоминающие ячейки энергонезависимой памяти обычно несимметричны по своей природе и позволяют записывать только нули (реже - только единицы) в предварительно стертые (чистые) ячейки. Однократно программируемые микросхемы позволяют изменять только исходное (после изготовления) состояние ячеек. Стирание ячеек выполняется либо для всей

микросхемы, либо для определенного блока, либо для одной ячейки (байта). Стирание приводит все биты стираемой области в одно состояние (обычно во все единицы, реже - во все нули). Процедура стирания обычно существенно дольше записи. В зависимости от способа стирания различают:

- микросхемы, стираемые ультрафиолетовым облучением, - их обычно называют просто *EPROM* (Eraseable ROM - стираемые микросхемы) или *UV-EPROM* (Ultra-Violet EPROM, УФПЗУ);
- электрически стираемые микросхемы *EEPROM* (Electrical Eraseable PROM, ЭСПЗУ), в том числе и флэш-память.

Процедура программирования многих типов памяти требует наличия относительно высокого напряжения программирования (12-26 В). Стирание или программирование микросхем может выполняться либо в специальном устройстве - программаторе, либо в самом целевом устройстве, если у него предусмотрены соответствующие средства. Для микросхем, не извлекаемых из целевого устройства (PC), возможны два метода их перепрограммирования:

- используя собственный процессор PC - *ISW* (In-System Write);
- подключая к плате внешний программатор - *OBP* (On-Board Programming).

Энергонезависимая память в основном применяется для хранения неизменяемой (или редко изменяемой) информации - системного программного обеспечения (BIOS), таблиц (например, знакогенераторов графических адаптеров), памяти конфигурации устройств. Эта информация обычно является ключевой для функционирования PC, поэтому весьма существенна забота о ее сохранности и предотвращении несанкционированного изменения. Нежелательное (ошибочное или под воздействием вируса) изменение содержимого становится возможным при использовании для хранения BIOS флэш-памяти, программируемой в целевом устройстве (на системной плате PC). Флэш-память используется и в качестве внешней памяти (как альтернатива дисковой), позволяющей как считывать, так и оперативно записывать новые данные.

Важными параметрами энергонезависимой памяти является время хранения и устойчивость к электромагнитным воздействиям, а для перепрограммируемой памяти еще и гарантированное количество циклов перепрограммирования.

Энергонезависимую память, запись в которую производят при регулярной работе, называют *NVRAM* (Non-Volatile Random Access Memory). Это название подразумевает возможность произвольной смены информации не только во всей ее области, но и в отдельной ячейке или небольшом блоке.

Программирование масочного ПЗУ производится на заводе изготовителе, что очень неудобно для мелких и средних серий производства, не говоря уже о стадии разработки устройства. Естественно, что для крупносерийного производства масочные ПЗУ являются самым дешевым видом ПЗУ. Для мелких и средних серий производства радиоаппаратуры были разработаны микросхемы, которые можно программировать в специальных устройствах - программаторах. В этих микросхемах постоянное соединение проводников в

запоминающей матрице заменяется плавкими перемычками, изготовленными из поликристаллического кремния. При производстве микросхемы изготавливаются все перемычки, что эквивалентно записи во все ячейки памяти логических единиц. В процессе программирования на выходы питания и выходы микросхемы подаётся повышенное питание. При этом, если на выход микросхемы подаётся напряжение питания (логическая единица), то через перемычку ток протекать не будет и перемычка останется неповрежденной. Если же на выход микросхемы подать низкий уровень напряжения (присоединить к корпусу), то через перемычку будет протекать ток, который испарит эту перемычку и при последующем считывании информации из этой ячейки будет считываться логический ноль.

Программируемые ПЗУ оказались очень удобны при мелкосерийном и среднесерийном производстве. Однако при разработке радиоэлектронных устройств часто приходится менять записываемую в ПЗУ программу. ПЗУ при этом невозможно использовать повторно, поэтому раз записанное ПЗУ при ошибочной или промежуточной программе приходится выкидывать, что естественно повышает стоимость разработки аппаратуры. Для устранения этого недостатка был разработан еще один вид ПЗУ, который мог бы стираться и программироваться заново.

Программирование **масочных ПЗУ** происходит в процессе изготовления БИС. Обычно на кристалле полупроводника вначале создаются все **запоминающие элементы (ЗЭ)**, а затем на заключительных технологических операциях с помощью фотошаблона слоя коммутации реализуются связи между линиями адреса, данных и собственно запоминающим элементом. Этот шаблон (маска) выполняется в соответствии с пожеланиями заказчика по картам заказа. Перечень возможных вариантов карт заказов приводится в технических условиях на ИМС ПЗУ. Такие ПЗУ изготавливаются на основе матриц диодов, биполярных или МОП-транзисторов.

Масочные ПЗУ на основе диодной матрицы

Схема такого ПЗУ представлена на рис. 1. Здесь горизонтальные линии – адресные, а вертикальные – это линии данных, с них в данном случае снимаются 8-разрядные двоичные числа. В данной схеме ЗЭ – это условное пересечение линии адреса и линии данных. Выбор всей строки ЗЭ производится при подаче логического нуля на линию адреса LA_i с соответствующего выхода дешифратора. В выбранный ЗЭ записывается логический 0 при наличии диода на пересечении линии D_i и LA_i , т.к. в этом случае замыкается цепь: + 5 В, диод, земля на адресной линии. Так, в данном ПЗУ при подаче адреса 11_2 активный нулевой сигнал появляется на адресной линии LA_3 , на ней будет уровень логического 0, на шине данных $D_7 \dots D_0$ появится информация 01100011_2 .

Масочные ПЗУ на основе матрицы МОП-транзисторов

Пример схемы данного ПЗУ представлен на рис. 2. Запись информации осуществляется подключением или неподключением МОП-транзистора в соответствующих точках БИС. При выборе определенного адреса на соответствующей адресной линии LA_i появляется активный сигнал логической

1, т.е. потенциал, близкий к потенциалу источника питания + 5 В. Данная логическая 1 подается на затворы всех транзисторов строки и открывает их. Если сток транзистора металлизирован, на соответствующей линии данных D_i появляется потенциал порядка 0,2 ... 0,3 В, т.е. уровень логического 0. Если же сток транзистора не металлизирован, указанная цепь не реализована, на сопротивлении R_i не будет падения напряжения, т.е. в точке D_i будет потенциал +5 В, т.е. уровень логической 1. Например, если в показанном на рис. 12.2 ПЗУ на адрес подать код 01_2 , на линии адреса $ЛА_1$ будет активный уровень 1, а на шине данных $D_3 \dots D_0$ будет код 0010_2 .

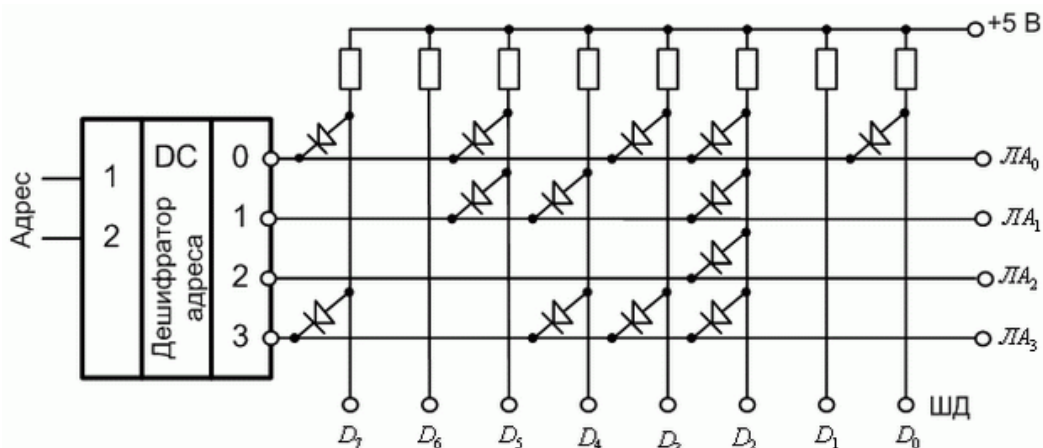


Рисунок 1 - Схема масочного ПЗУ на основе диодной матриц

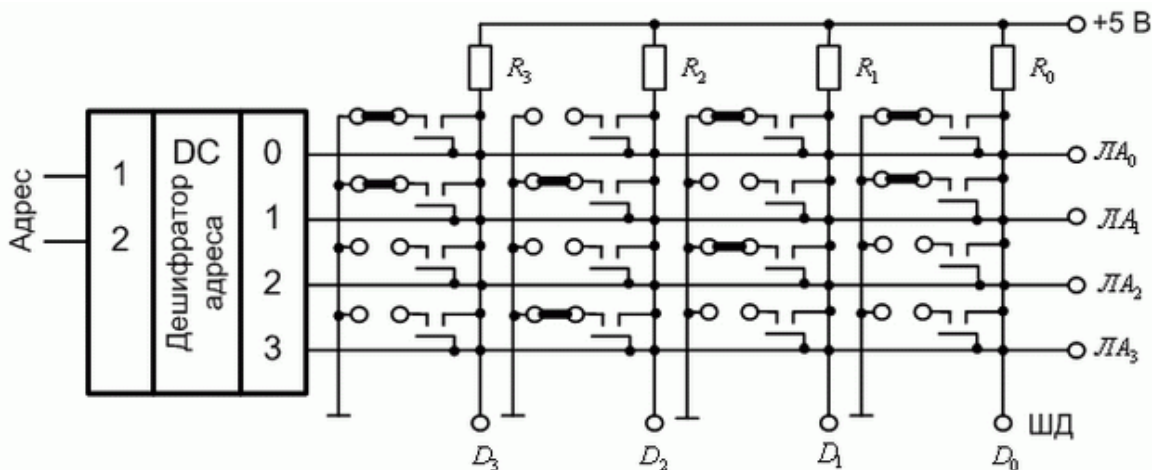


Рисунок 2 - Схема масочного ПЗУ на основе матрицы МОП-транзисторов

Масочные ПЗУ на основе матрицы биполярных транзисторов

Пример схемы данного ПЗУ представлен на рис. 3. Запись информации осуществляется также металлизацией или неметаллизацией участка между базой и адресной линией. Для выбора строки $3Э$ на линию адреса $ЛА_i$ подается логическая 1. При металлизации она подается на базу транзистора, он открывается вследствие разницы потенциалов между эмиттером (земля) и базой (примерно + 5 В). При этом замыкается цепь: + 5 В; сопротивление R_i ;

Таблица 1 - Параметры масочных ПЗУ			
Обозначение БИС	Технология изготовления	Информационная емкость, бит	Время выборки, нс
505PE3	pМОП	512x8	1500
K555PE4	ТТЛШ	2Кx8	800
K568PE1	nМОП	2Кx8	120
K596PE1	ТТЛ	8Кx8	350

Программируемые ПЗУ (ППЗУ) представляют собой такие же диодные или транзисторные матрицы, как и масочные ПЗУ, но с иным исполнением ЗЭ. Запоминающий элемент **ППЗУ** приведен на рис. 6. Доступ к нему обеспечивается подачей логического 0 на линию адреса $ЛА_i$. Запись в него производится в результате осаждения (расплавления) плавких вставок ПВ, включенных последовательно с диодами, эмиттерами биполярных транзисторов, стоками МОП-транзисторов. Плавкая вставка ПВ представляет собой небольшой участок металлизации, который разрушается (расплавляется) при программировании импульсами тока величиной 50 ... 100 микроампер и длительностью порядка 2 миллисекунд. Если вставка сохранена, то в ЗЭ записан логический 0, поскольку реализована цепь между источником питания и землей на $ЛА_i$ через диод (в транзисторных матрицах – через открытый транзистор). Если вставка разрушена, то указанной цепи, нет и в ЗЭ записана логическая 1.

На рис. 7 приведены примеры функциональных обозначений **ППЗУ**, выполненных по различным технологиям, а в табл. 2 – их основные параметры.

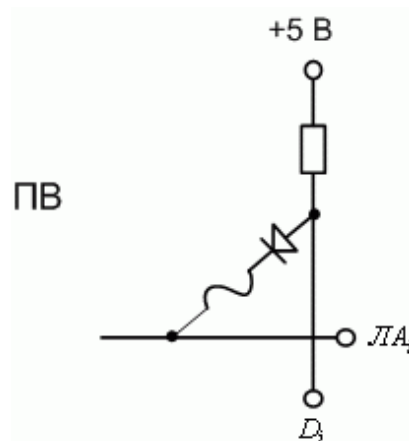


Рисунок 6 - Схема масочного ПЗУ на основе матрицы биполярных транзисторов

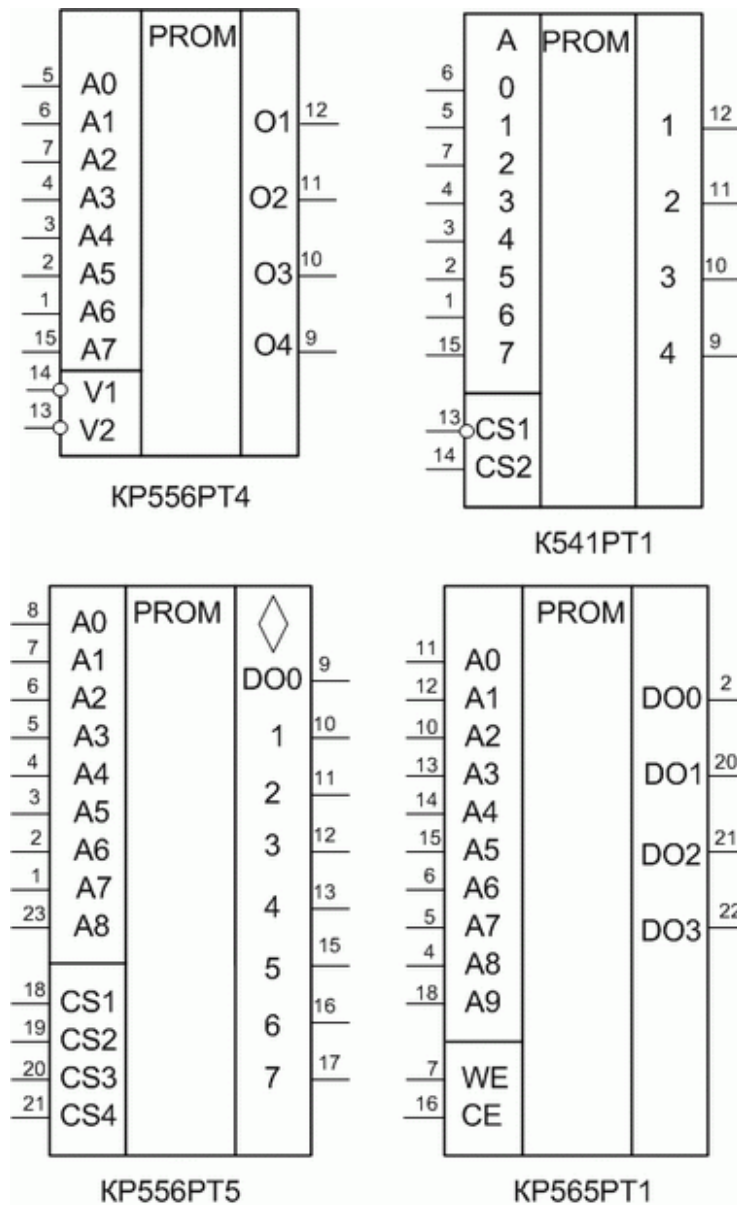


Рисунок 7 - Функциональные обозначения ППЗУ

Таблица 2 - Параметры ППЗУ

Обозначение БИС	Технология изготовления	Информационная емкость, бит	Время выборки, нс
KP556PT4	ТТЛШ	256x4	70
KP556PT5	ТТЛШ	512x8	70
K541PT1	И ² Л	256x4	80
KP565PT1	<i>n</i> МОП	1Кx4	300

ПЗУ с ультрафиолетовым стиранием строится на основе запоминающей матрицы построенной на ячейках памяти, внутреннее устройство которой приведено на следующем рисунке:

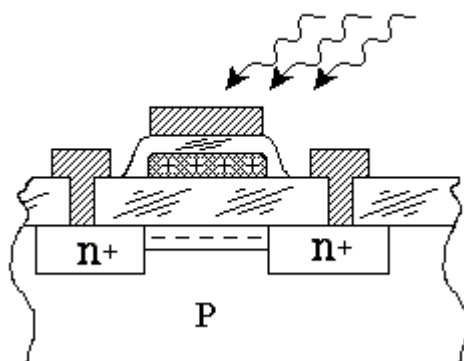


Рисунок 8 - Запоминающая ячейка ПЗУ с ультрафиолетовым и электрическим стиранием.

Ячейка представляет собой МОП транзистор, в котором затвор выполняется из поликристаллического кремния. Затем в процессе изготовления микросхемы этот затвор окисляется и в результате он будет окружен оксидом кремния - диэлектриком с прекрасными изолирующими свойствами. В описанной ячейке при полностью стертом ПЗУ заряда в плавающем затворе нет, и поэтому транзистор ток не проводит. При программировании микросхемы на второй затвор, находящийся над плавающим затвором, подается высокое напряжение и в плавающий затвор за счет туннельного эффекта индуцируются заряды. После снятия программирующего напряжения на плавающем затворе индуцированный заряд остается и, следовательно, транзистор остается в проводящем состоянии. Заряд на плавающем затворе может храниться десятки лет.

Структурная схема постоянного запоминающего устройства не отличается от описанного ранее масочного ПЗУ. Единственно вместо переключки используется описанная выше ячейка. В репрограммируемых ПЗУ стирание ранее записанной информации осуществляется ультрафиолетовым излучением. Для того, чтобы этот свет мог беспрепятственно проходить к полупроводниковому кристаллу, в корпус микросхемы встраивается окошко из кварцевого стекла.

При облучении микросхемы, изолирующие свойства оксида кремния теряются и накопленный заряд из плавающего затвора стекает в объем полупроводника и транзистор запоминающей ячейки переходит в закрытое состояние. Время стирания микросхемы колеблется в пределах 10 - 30 минут.

Количество циклов записи - стирания микросхем находится в диапазоне от 10 до 100 раз, после чего микросхема выходит из строя. Это связано с разрушающим воздействием ультрафиолетового излучения. В этих микросхемах раньше хранились программы BIOS универсальных компьютеров.

В такой микросхеме имеется небольшое окошко с кварцевым стеклом. За ним находится кристалл, который облучается ультрафиолетом. После стирания информации это окошко заклеивают. Частичная перезапись данных по-прежнему остается невозможной, так как рентгеновские и ультрафиолетовые

лучи изменяют все биты стираемой области в положение 1. Повторная запись данных осуществляется также на программаторах (как в ROM и EPROM). Вообще, EPROM была основана на МОП (металл-оксид-полупроводник) транзисторах. Запись данных в ячейки такого транзистора производилась методом лавинной инжекции заряда (о методах записи будет сказано ниже). Этот метод давал возможность неоднократно перезаписывать данные памяти (хотя количество циклов было ограниченным).

2 Флэш-память

Флэш-память по определению относится к классу EEPROM, но использует особую технологию построения запоминающих ячеек. Стирание во флэш-памяти производится сразу для целой области ячеек (блоками или полностью всей микросхемы). Это позволило существенно повысить производительность в режиме записи (программирования). Флэш-память обладает сочетанием высокой плотности упаковки (ее ячейки на 30% меньше ячеек DRAM), энергонезависимого хранения, электрического стирания и записи, низкого потребления, высокой надежности и невысокой стоимости.

Современная флэш-память имеет время доступа при чтении 35-200 нс, существуют версии с интерфейсом динамической памяти и ... напоминающим интерфейс синхронной статической памяти. Стирание информации (поблочное или во всей микросхеме) занимает 1-2 сек. Программирование (запись) байта занимает время порядка 10 мкс, причем шинные циклы при записи - нормальные для процессора (не растянутые, как для EPROM и EEPROM).

Интерфейс микросхем флэш-памяти хорошо сочетается со стандартными сигналами, используемыми в микропроцессорных системах. Внутренние циклы стирания, записи и верификации (проверки) выполняются автономно от шинных циклов внешнего интерфейса, что является существенным преимуществом перед микросхемами EPROM и EEPROM.

Первые микросхемы флэш-памяти были предложены фирмой Intel в 1988 году и с тех пор претерпели существенные изменения по архитектуре, интерфейсу и напряжению питания. По организации массива различаются микросхемы:

- *Bulk Erase* - стирание возможно только для всего объема.
- *Boot Lock* - массив разделен на несколько блоков разного размера, стираемых независимо. Один из блоков имеет дополнительные аппаратные средства защиты от стирания и записи.
- *Flash File* - массив разделен на несколько равноправных независимо стираемых блоков обычно одинакового размера, что позволяет их называть микросхемами с симметричной архитектурой SA (Symmetrical Architecture).

Микросхемы Boot Lock специально предназначены для хранения BIOS, а привелигерованный блок (Boot Block) хранит минимальный загрузчик, позволяющий загрузить (например с дискеты) и выполнить утилиту программирования основного блока флэш-памяти. В обозначении этих микросхем присутствует суффикс T (Top) или B (Bottom), определяющий

положение Boot-блока либо в старших, либо в младших адресах соответственно. Первые предназначены для процессоров, стартующих со старших адресов (в том числе x86, Pentium), вторые - для стартующих с нулевого адреса.

Микросхемы с небольшим размером стираемого сектора кроме BIOS в блоках параметров могут хранить и конфигурационную информацию (ESCD системы PnP, конфигурацию устройств EISA и MCA). Применение микросхем большого объема позволяет кроме BIOS хранить и дополнительный резидентный код, например, в портативных компьютерах во флэш-память может помещаться ядро ОС (MS-DOS), что позволяет экономить энергию за счет сокращения количества обращений к диску.

Гибкость системы, обретаемая при использовании флэш-памяти как носителя BIOS, имеет и негативные обратные стороны - возможность повреждения в случае неудачной записи или записи неподходящей версии и появление новой и достаточно благодатной почвы для вирусов, которые могут незаметно переписать код BIOS в своих диверсионных целях. В связи с этим актуальна *защита BIOS от несанкционированного изменения*.

Микросхемы флэш-памяти выпускаются многими фирмами. Они различаются по организации, интерфейсу, напряжению питания и программирования, методам защиты и другим параметрам. Лидеры в области разработки и производства флэш-памяти - фирмы AMD, Fujitsu Corporation, Intel Corporation и Sharp Corporation летом 1996 года приняли спецификацию CFI (Common Flash Interface), обеспечивающую совместимость разрабатываемого программного обеспечения с существующими и разрабатываемыми моделями флэш-памяти.

По устройству чип флэш-памяти отдаленно напоминает микросхему динамической энергозависимой памяти, только вместо конденсаторов в ячейках памяти установлены полупроводниковые приборы — транзисторы. При подаче напряжения на выводы транзистора он принимает одно из фиксированных положений — закрытое или открытое. И остается в этом положении до тех пор, пока на выводы транзистора не будет подан электрический заряд, изменяющий его состояние. Таким образом, последовательность логических нулей и единиц формируется в этом типе памяти подобно ПЗУ — закрытые для прохождения электрического тока ячейки распознаются как логические единицы, открытые — как логические нули.

Таким образом, в самом простом случае ячейка Flash состоит из одного полевого транзистора. Элемент включает в себя специальную электрически изолированную область, называемую «плавающим затвором». Этот термин возник из-за того, что потенциал этой области не является стабильным, что позволяет накапливать в ней электроны (именно здесь и хранится вся информация памяти). Выше «плавающего» находится управляющий затвор, который является неотъемлемой частью при процессе записи/стирания данных памяти. Эта область напрямую соединена с линией слов. Перпендикулярно этой линии располагается линия битов, которая соединена со стоком (при записи данных из этой области транзистора появляется поток электронов). Сток

разделяется с истоком специальной подложкой, которая не проводит электрический ток.

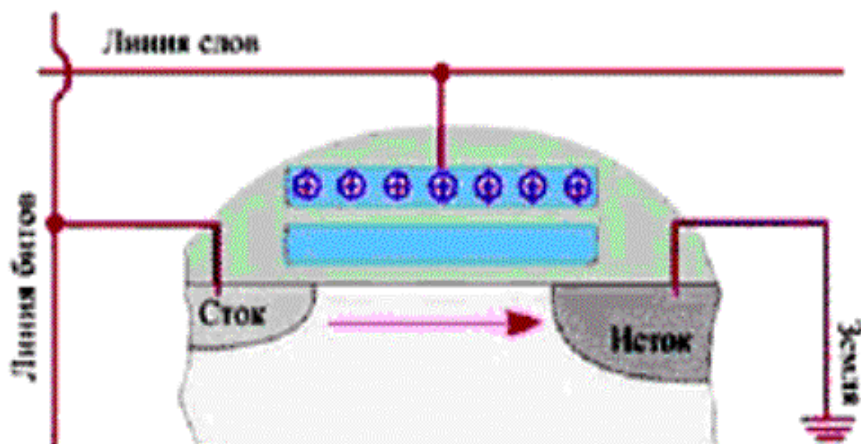


Рисунок 9 - Ячейка Flash памяти

В начале развития Flash каждая ячейка памяти хранила один бит информации и состояла из одного полевого транзистора. Прогресс не стоит на месте, через несколько лет после выпуска чудо-микросхемы были проведены успешные испытания Flash, в которых ячейка хранила уже два бита. Естественно, что на такую память можно было записать в два раза больше информации. В настоящее время уже существуют теоретические разработки памяти с четырехбитными ячейками.

Как же устроена такая ячейка? Ведь теоретически наличие заряда в ячейке памяти означает 1, отсутствие 0, остальные значения представить невозможно. Но на самом деле, в микросхеме существует различие величин заряда, которые накапливаются на «плавающем» затворе. Благодаря этому различию, информация в ячейке может быть представлена различными битовыми комбинациями. Величину заряда на затворе можно определить измерением порогового (максимального) напряжения транзистора и по итогам этого измерения представить битовую комбинацию.

Перезапись и стирание Flash значительно изнашивает микросхему, поэтому технологии производства памяти постоянно совершенствуются, внедряются оптимизирующие способы записи микросхемы, а также алгоритмы, направленные на равномерное использование всех ячеек в процессе работы.

Преимущества флэш-памяти в независимости от наличия или отсутствия электрического питания, в долговременности хранения информации (производители гарантируют сохранность данных на протяжении 10 лет, но на практике должно быть больше) и в высокой механической надежности (в накопителях на базе флэш-памяти нет никаких механических устройств, следовательно, нечему ломаться). Недостатки - в высокой сложности устройства (транзисторы имеют микронные размеры), в невысоком быстродействии (время изменения состояния транзистора больше, чем время заряда-разряда конденсатора) и в относительно высокой стоимости микросхем.

Флэш-память быстро прогрессирует. За последние несколько лет появились новые типы микросхем - был осуществлен массовый переход с 5-вольтовой технологии питания на 3,3-вольтовую, были применены новые типы полупроводниковых приборов, разработаны и внедрены в производство механизмы ускорения процедуры записи-чтения информации. Хотя Flash и лидирует на компьютерном рынке, ее могут вытеснить другие новые технологии. Например, новейшая память на кремниевых нанокристаллах. Отличие такой памяти от Flash в следующем: подложка между стоком и истоком состоит из кремниевых нанокристалльных сфер. Такая прослойка предотвращает передачу заряда с одного нанокристалла на другой, повышая, таким образом, надежность - один дефект не ведет к полному сбою, как в нынешней энергонезависимой памяти на транзисторах с плавающим затвором. Первый в мире работоспособный образец такой памяти был предоставлен компанией Motorola.

Ячейки флэш-памяти бывают как на одном, так и на двух транзисторах. В простейшем случае каждая ячейка хранит один бит информации и состоит из одного полевого транзистора со специальной электрически изолированной областью ("плавающим" затвором - floating gate), способной хранить заряд многие годы. Наличие или отсутствие заряда кодирует один бит информации. При записи заряд помещается на плавающий затвор одним из двух способов (зависит от типа ячейки): методом инъекции "горячих" электронов или методом туннелирования электронов. Стирание содержимого ячейки (снятие заряда с "плавающего" затвора) производится методом туннелирования. Как правило, наличие заряда на транзисторе понимается как логический "0", а его отсутствие - как логическая "1". Современная флэш-память (2006 г.) обычно изготавливается по 0,13- и 0,18-микронному техпроцессу.

Принцип чтения микросхемы Flash довольно прост и базируется на законах квантовой механики. При извлечении данных из памяти, заряд на "плавающем" затворе отсутствует, а на управляющий затвор подается заряд положительного направления. Под его воздействием между стоком и истоком создается канал трассировки (свободная зона на кристалле транзистора, выделенная для реализации межсоединений ячеек). Все это происходит за счет туннельного эффекта, а данные памяти затем можно считывать с истока.

Если на "плавающем" затворе имеется заряд, то обычного напряжения (которое подается при чтении) недостаточно. Поэтому при записи применяют метод инъекции электронов. Суть его заключается в следующем: на управляющий затвор и исток подается высокое напряжение (причем на затворе оно в два раза выше). Благодаря этому напряжению, электроны способны преодолеть тонкую пленку диэлектрика и попасть на "плавающий" затвор. Такой процесс получил название "инъекция горячих электронов" (термин "горячий" условен, электроны были названы так, потому что обладают высокой энергией, достаточной для преодоления слоя диэлектрика).

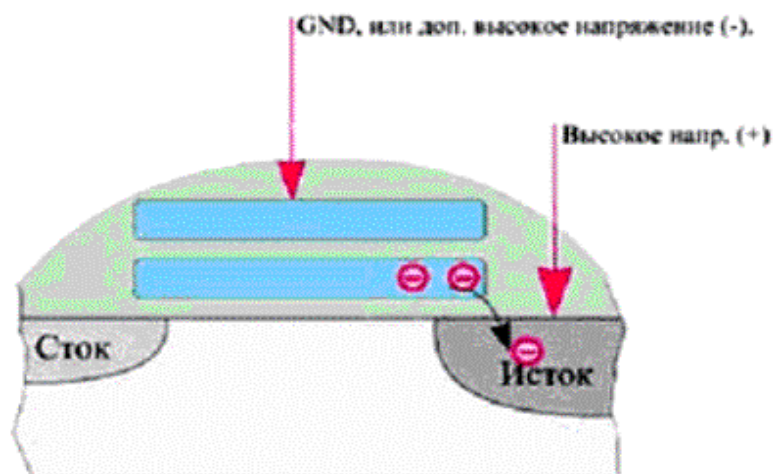


Рисунок 10 - Чтение данных из ячейки памяти

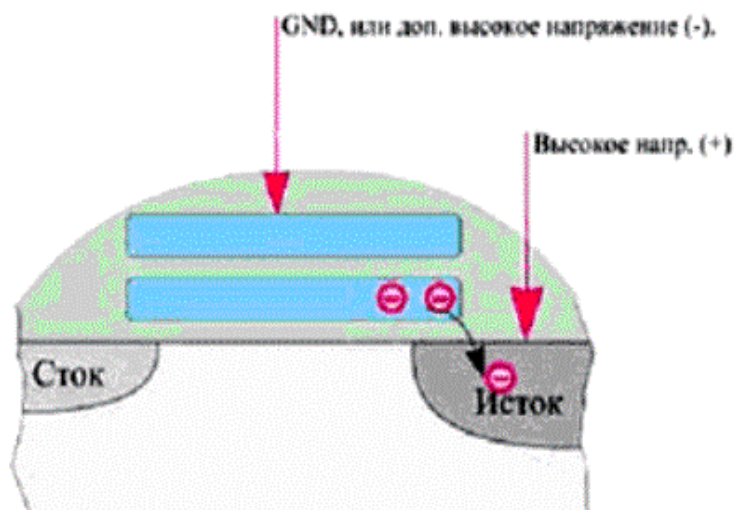


Рисунок 11 - Запись данных в ячейку памяти

Чтобы стереть информацию из памяти, достаточно подать высокое положительное напряжение на исток. Под его воздействием отрицательные электроны с “плавающего” затвора (благодаря туннельному эффекту) переходят в область истока. Процесс продолжается до полной разрядки затвора. Ускорить метод туннелирования электронов можно путем подачи дополнительного высокого отрицательного напряжения на управляющий затвор.

Эффект туннелирования - один из эффектов, использующих волновые свойства электрона. Сам эффект заключается в преодолении электроном потенциального барьера малой "толщины". Для наглядности представим себе структуру, состоящую из двух проводящих областей, разделенных тонким слоем диэлектрика (обеднённая область). Преодолеть этот слой обычным способом электрон не может - не хватает энергии. Но при создании определённых условий (соответствующее напряжение и т.п.) электрон проскакивает слой диэлектрика (туннелирует сквозь него), создавая ток. Мы

рассмотрели простейший случай, когда каждая ячейка Flash хранит один бит информации и состоит из одного полевого транзистора.

3 Память с последовательным доступом

Устройства памяти с последовательным доступом позволяют осуществлять доступ к данным последовательно, т.е. для того, чтобы считать нужный блок памяти, необходимо считать все предшествующие блоки. Среди устройств памяти с последовательным доступом выделяют:

1. Накопители на магнитных лентах (НМЛ) – устройства считывания данных с магнитной ленты. Такие накопители достаточно медленные, хотя и большой ёмкости. Современные устройства для работы с магнитными лентами – **стримеры** – имеют увеличенную скорость записи 4 - 5Мбайт в сек. Существуют также, устройства позволяющие записывать цифровую информацию на видеокассеты, что позволяет хранить на 1 кассете 2 Гбайта информации. Магнитные ленты обычно используются для создания архивов данных для долговременного хранения информации.

2. Перфокарты – карточки из плотной бумаги и **перфоленты** – катушки с бумажной лентой, на которых информация кодируется путем пробивания (перфорирования) отверстий. Для считывания данных применяются устройства последовательного доступа. В настоящее время данные устройства морально устарели и не применяются.

Различные виды памяти имеют свои достоинства и недостатки. Так, внутренняя память имеет хорошее быстродействие, но ограниченный объем. Внешняя память, наоборот, имеет низкое быстродействие, но неограниченный объем. Производителям и пользователям компьютеров приходится искать компромисс между объемом памяти, скоростью доступа и ценой компьютера, так комбинируя разные виды памяти, чтобы компьютер работал оптимально. В любом случае, объем оперативной памяти является основной характеристикой ЭВМ и определяет производительность компьютера.

4 Видеопамять

Видеопамять — часть оперативной памяти, отведённая для хранения данных, которые используются для формирования изображения на экране монитора.

При этом в видеопамяти может содержаться как непосредственно растровый образ изображения (экранный кадр), так и отдельные фрагменты как в растровой (текстуры), так и в векторной (многоугольники, в частности треугольники) формах.

Существует выделенная оперативная память видеокарты, также называемая «**видеопамять**». Такая оперативная память используется только под нужды различных графических приложений и игр.

Как правило, чипы оперативной памяти современной видеокарты припаяны прямо к текстолиту печатной платы, в отличие от съёмных модулей

системной памяти, которые вставляются в стандартизированные разъёмы ранних видеоадаптеров.

При изготовлении видеокарт уже достаточно давно используется память GDDR3. На смену ей пришла GDDR4, которая имеет более высокую пропускную способность, чем GDDR3; однако GDDR4 не получила широкого распространения вследствие плохого соотношения «Цена-производительность» и ограниченно использовалась лишь в некоторых топовых видеокартах (например Radeon X1950XTX, HD 2900 XT, HD3870). Далее появилась память GDDR5, которая по состоянию на 2012 год является наиболее массовой, GDDR3 используется в бюджетном сегменте.

Так же, видеопамять отличается от «обычной» системной ОЗУ более жёсткими требованиями к ширине шины.

Шина памяти – это магистраль, связывающая графический процессор и память видеокарт.

Шина данных видеопамяти бывает:

- 32-битной,
- 64-битной,
- 128-битной,
- 192-битной (нестандартная шина памяти),
- 256-битной,
- 320-битной (нестандартная шина памяти),
- 384-битной (нестандартная шина памяти),
- 448-битной (нестандартная шина памяти),
- 512-битной.

Имеет значение пропорциональность количества памяти, её типа и ширина шины данных: 512 МБ DDR2, при ширине шины данных в 128 бит, будет работать медленнее и гораздо менее эффективно, чем 256 МБ GDDR3 при ширине шины в 128 бит и т.п. По понятным причинам, 256 МБ GDDR3 с шириной шины 256 бит лучше, чем 256 МБ GDDR3 с шириной шины в 128 бит и т.п.

Также стоит учитывать, что из-за относительно невысокой стоимости видеопамяти многие производители видеокарт устанавливают избыточное количество видеопамяти (1-2 Гбайт) на слабые видеокарты с целью повышения их маркетинговой привлекательности.

Требования современных операционных систем и компьютерных игр возрастают; так, например чтобы играть комфортно в наиболее современные игры:

- на период 2008—2009 года требовалось порядка 512 МБ (и более) GDDR3 256 бит или 2 ГБ (и более) DDR2(3),
- на период 2010 года требовалось порядка 768 МБ (и более) GDDR4 320 бит или 3 ГБ (и более) DDR3,
- на период 2011 года требуется порядка 1024 МБ (и более) GDDR5 256 бит или 3-4 ГБ (и более) DDR3.

5 Буферная память типа FIFO

В вычислительных системах используются подсистемы с различным быстродействием, и, в частности, с различной скоростью передачи данных (рис. 12). Обычно обмен данными между такими подсистемами реализуется с использованием прерываний или канала прямого доступа к памяти. В первую очередь подсистема 1 формирует запрос на обслуживание по мере готовности данных к обмену. Однако обслуживание прерываний связано с непроизводительными потерями времени и при пакетном обмене производительность подсистемы 2 заметно уменьшается. При обмене данными с использованием канала прямого доступа к памяти подсистема 1 передает данные в память подсистемы 2. Данный способ обмена достаточно эффективен с точки зрения быстродействия, но для его реализации необходим довольно сложный контроллер прямого доступа к памяти.



Рисунок 12 - Применение буферной памяти.

Наиболее эффективно обмен данными между подсистемами с различным быстродействием реализуется при наличии между ними специальной буферной памяти. Данные от подсистемы 1 временно запоминаются в буферной памяти до готовности подсистемы 2 принять их. Емкость буферной памяти должна быть достаточной для хранения тех блоков данных, которые подсистема 1 формирует между считываниями их подсистемой 2. Отличительной особенностью буферной памяти является запись данных с быстродействием и под управлением подсистемы 1, а считывание - с быстродействием и под управлением подсистемы 2 ("эластичная память"). В общем случае память должна выполнять операции записи и считывания совершенно независимо и даже одновременно, что устраняет необходимость синхронизации подсистем. Буферная память должна сохранять порядок поступления данных от подсистемы 1, т.е. работать по принципу "первое записанное слово считывается первым" (First Input First Output - FIFO). Таким образом, под буферной памятью типа FIFO понимается ЗУПВ, которое автоматически следит за порядком поступления данных и выдает их в том же порядке, допуская выполнение независимых и одновременных операций записи и считывания. На рис. 13 приведена структурная схема буферной памяти типа FIFO емкостью 64x4.

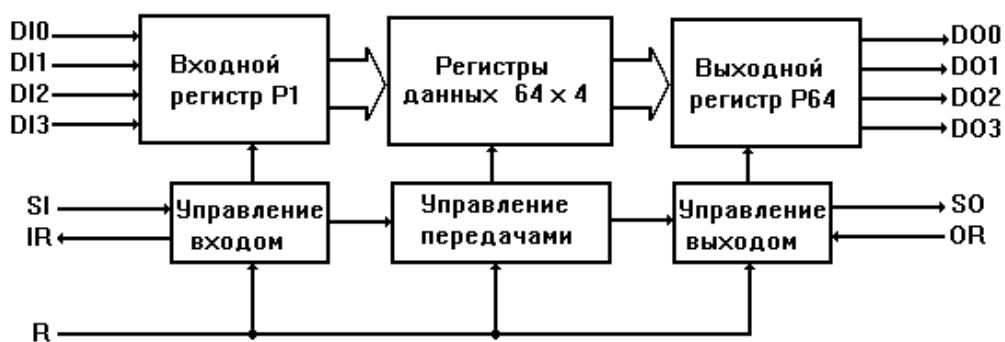


Рисунок 13 - Структурная схема буфера 64x4

На кристалле размещены 64 4-битных регистра с независимыми цепями сдвига, организованных в 4 последовательных 64-битных регистра данных, 64-битный управляющий регистр, а также схема управления. Входные данные поступают на линии DI0-DI3, а вывод данных осуществляется через контакты DO0-DO3. Ввод (запись) данных производится управляющим сигналом SI (shift in), а вывод (считывание) - сигналом вывода SO (shift out). Ввод данных осуществляется только при наличии сигнала готовности ввода IR (input ready), а вывод - при наличии сигнала готовности вывода OR (output ready). Управляющий сигнал R (reset) производит сброс содержимого буфера.

При вводе 4-битного слова под действием сигнала SI оно автоматически передвигается в ближайший к выходу свободный регистр. Состояние регистра данных отображается в соответствующем ему управляющем триггере, совокупность триггеров образует 64-битный управляющий регистр. Если регистр содержит данные, то управляющий триггер находится в состоянии 1, а если регистр не содержит данных, то триггер находится в состоянии 0. Как только управляющий бит соседнего справа регистра изменяется на 0, слово данных автоматически сдвигается к выходу. Перед началом работы в буфер подается сигнал сброса R и все управляющие триггеры переводятся в состояние 0 (все регистры буфера свободны). На выводе IR формируется логическая 1, т.е. буфер готов воспринимать входные данные. При действии сигнала ввода SI входное слово загружается в регистр P1, а управляющий триггер этого регистра устанавливается в состояние 1: на входе IR формируется логический 0. Связи между регистрами организованы таким образом, что поступившее в P1 слово "спонтанно" копируется во всех регистрах данных FIFO и появляется на выходных линиях DO0-DO3. Теперь все 64 регистра буфера содержат одинаковые слова, управляющий триггер последнего регистра P64 находится в состоянии 1, а остальные управляющие триггеры сброшены при передаче данных в соседние справа регистры. Состояние управляющего триггера P64 выведено на линию готовности выхода OR; OR принимает значение 1, когда в триггер записывается 1. Процесс ввода может продолжаться до полного заполнения буфера; в этом случае все

управляющие триггеры находятся в состоянии 1 и на линии IR сохраняется логический 0.

При подаче сигнала SO производится восприятие слова с линий DO0-DO3, управляющий триггер P64 переводится в состояние 1, на линии OR появляется логическая 1, а управляющий триггер P64 сбрасывается в 0. Затем этот процесс повторяется для остальных регистров и нуль в управляющем регистре перемещается ко входу по мере сдвига данных вправо.

В некоторых кристаллах буфера FIFO имеется дополнительная выходная линия флажка заполнения наполовину. На ней формируется сигнал 1, если число слов составляет более половины емкости буфера.

Рассмотренный принцип организации FIFO допускает выполнение записи и считывания данных независимо и одновременно. Скорость ввода определяется временным интервалом, необходимым для передачи данных из P1, а выводить данные можно с такой же скоростью. Единственным ограничением является время распространения данных через FIFO, равное времени передачи входного слова на выход незаполненного буфера FIFO. Оно равняется произведению времени внутреннего сдвига и числа регистра данных. В буферах FIFO, выполненных по МОП-технологии и имеющих емкость 64 слова, время распространения составляет примерно 30 мкс, а в биполярных FIFO такой же емкости - примерно 2 мкс.

Буферы можно наращивать как по числу слов, так и по их длине.

6 Стековая память (LIFO)

Стековой называют память, доступ к которой организован по принципу: "последним записан - первым считан" (Last Input First Output - LIFO). Использование принципа доступа к памяти на основе механизма LIFO началось с больших ЭВМ. Применение стековой памяти оказалось очень эффективным при построении компилирующих и интерпретирующих программ, при вычислении арифметических выражений с использованием польской инверсной записи. В малых ЭВМ она стала широко использоваться в связи с удобствами реализации процедур вызова подпрограмм и при обработке прерываний.

Принцип работы стековой памяти состоит в следующем (см. рис. 14). Когда слово А помещается в стек, оно располагается в первой свободной ячейке памяти. Следующее записываемое слово перемещает предыдущее на одну ячейку вверх и занимает его место и т.д. Запись 8-го слова, после Н, приводит к переполнению стека и потере слова А. Считывание слов из стека осуществляется в обратном порядке, начиная с слова Н, который был записан последним. Заметим, что выборка, например, слова Е невозможна до выборки слова F, что определяется механизмом обращения при записи и чтении типа LIFO. Для фиксации переполнения стека желательно формировать признак переполнения.

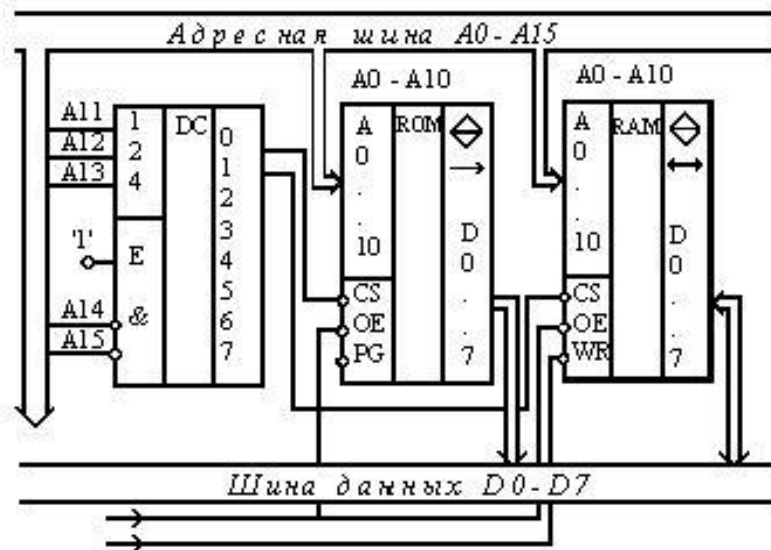


Рисунок 14 - Принцип работы стековой памяти

Перемещение данных при записи и считывании информации в стековой памяти подобно тому, как это имеет место в сдвигающих регистрах. С точки зрения реализации механизма доступа к стековой памяти выделяют аппаратный и аппаратно-программный (внешний) стеки. Аппаратный стек представляет собой совокупность регистров, связи между которыми организованы таким образом, что при записи и считывании данных содержимое стека автоматически сдвигается. Обычно емкость аппаратного стека ограничена диапазоном от нескольких регистров до нескольких десятков регистров, поэтому в большинстве МП такой стек используется для хранения содержимого программного счетчика и его называют стеком команд. Основное достоинство аппаратного стека - высокое быстродействие, а недостаток - ограниченная емкость.

Наиболее распространенным в настоящее время и, возможно, лучшим вариантом организации стека в ЭВМ является использование области памяти. Для адресации стека используется указатель стека, который предварительно загружается в регистр и определяет адрес последней занятой ячейки. Помимо команд CALL и RET, по которым записывается в стек и восстанавливается содержимое программного счетчика, имеются команды PUSH и POP, которые используются для временного запоминания в стеке содержимого регистров и их восстановления, соответственно. В некоторых МП содержимое основных регистров запоминается в стеке автоматически при прерывании программ. Содержимое регистра указателя стека при записи уменьшается, а при считывании увеличивается на 1 при выполнении команд PUSH и POP, соответственно.

Контрольные вопросы:

1. Какие существуют типы энергонезависимой памяти?
2. Режимы работы постоянной памяти ROM.
3. Какие различают виды постоянной памяти.
4. Охарактеризуйте масочные ПЗУ типа ROM.
5. Какие используются основные схемы построения масочных ПЗУ.
6. Что представляют собой программируемые ПУЗ?
7. Укажите основные характеристики репрограммируемых ПЗУ со стиранием ультрафиолетовыми лучами (EPROM) и электрическим стиранием (EEPROM).
8. Поясните назначение видеопамати.
9. Для чего используется буферная память?
10. Как работает стековая память?
11. К какому классу памяти относится флэш-память?
12. Охарактеризуйте особенности флэш-памяти.

Тема 7. Процессоры ЭВМ

1. Процессорные устройства. Структура микропроцессорной системы.
2. Основные блоки процессора.
3. Работа процессора. Организация обрабатываемой части микропроцессора.
4. RISC – процессоры с сокращенной системой команд и CISC (Complete Instruction Set Computer) процессоры с полной системой команд.

1 Процессорные устройства. Структура микропроцессорной системы

Важнейший компонент любого персонального - это его микропроцессор. Данный элемент в большей степени определяет возможности вычислительной системы и, образно выражаясь, является его сердцем. До настоящего времени безусловным лидером в создании современных микропроцессоров остаётся фирма Intel.

МП, как правило, представляет собой сверхбольшую интегральную схему, реализованную в едином полупроводниковом кристалле и способную выполнять функции центрального процессора. Степень интеграции определяется размерами кристалла и количеством реализованных в нём транзисторов. Часто интегральными микросхемы называют чипами (chips).

К обязательным компонентам микропроцессора относятся арифметико-логическое (исполнительное) устройство и блок управления. Они характеризуются скоростью (тактовой частотой), разрядностью или длиной слова (внутренней и внешней), архитектурой и набором команд. Архитектура микропроцессора определяет необходимые регистры, стеки, систему адресации, а также типы обрабатываемых процессором данных. Выполняемые микропроцессором команды предусматривают, как правило, арифметические действия, логические операции, передачу управления (условную и безусловную) и перемещение данных (между регистрами, памятью, портами ввода-вывода).

Под конвейерным режимом понимают такой вид обработки, при котором интервал времени, требуемый для выполнения процесса в функциональном узле (например, в арифметико-логическом устройстве) микропроцессора, продолжительнее, чем интервалы, через которые данные могут вводиться в этот узел. Предполагается, что функциональный узел выполняет процесс в несколько этапов, то есть когда первый этап завершается, результаты передаются на второй этап, на котором используются другие аппаратные средства. Разумеется, что устройство, используемое на первом этапе, оказывается свободным для начала новой обработки данных. Как известно, можно выделить четыре этапа обработки команды микропроцессора: выборка, декодирование, выполнение и запись результата. Иными словами, в ряде случаев пока первая команда выполняется, вторая может декодироваться, а третья выбираться.

С внешними устройствами микропроцессор может «общаться» благодаря шинам адреса, данных и управления, выведенных на специальные контакты корпуса микросхемы. Стоит отметить, что разрядность внутренних регистров микропроцессора может не совпадать с количеством внешних выводов для линий данных. Иначе говоря, микропроцессор с 32-разрядными регистрами может иметь, например, только 16 линий внешних данных. Объём физически адресуемой микропроцессорной памяти однозначно определяется разрядностью внешней шины адреса как 2^N , где N - количество адресных линий.

Любое внешнее устройство, совершающее по отношению к микропроцессору операции ввода-вывода, можно назвать **периферийным**.

Регистр представляет собой совокупность бистабильных устройств (то есть имеющих два устойчивых состояния), предназначенных для хранения информации и быстрого доступа к ней. В качестве таких устройств в интегральных схемах используют триггеры. Триггер в свою очередь выполнен на транзисторных переключателях (электронных ключах). В регистре из N триггеров можно запомнить слово из N бит информации.

Порт - это некая схема сопряжения, обычно включающая в себя один или несколько регистров ввода-вывода и позволяющая подключить, например периферийное устройство к внешним шинам микропроцессора. Практически каждая микросхема использует для различных целей несколько портов ввода-вывода. Каждый порт на персональном компьютере имеет свой уникальный номер. Заметим, что номера порта - это, по сути, адрес регистра ввода-вывода, причём адресные пространства основной памяти и портов ввода-вывода не пересекаются.

Под прерыванием понимается сигнал, по которому процессор узнаёт о совершении некоторого асинхронного события. При этом исполнение текущей последовательности команд приостанавливается (прерывается), а в место неё начинает выполняться другая последовательность, соответствующая данному прерыванию. Прерывания можно классифицировать как аппаратные, логические и программные. Аппаратные прерывания обычно связаны с запросами от периферийных устройств (например, нажатие клавиши клавиатуры), логические возникают при работе самого МП (деление на ноль), а программные инициализируются выполняемой программой и используются для вызова специальных подпрограмм. Кроме того, прерывания могут быть маскируемыми, то есть при определённых условиях (например, запрете на определение прерывания) микропроцессор не обращает на них внимание, и немаскируемыми. В последнем случае, как правило, должны обрабатываться почти катастрофические события (падение напряжения питания или ошибка памяти).

В режиме прямого доступа (DMA, Direct Memory Access) периферийное устройство связано с оперативной памятью непосредственно, минуя внутренние регистры микропроцессора. Наиболее эффективна такая передача данных в ситуациях, когда требуется высокая скорость обмена при передаче

большого количества информации (например, при загрузке данных в память с внешнего накопителя).

Архитектура типичной небольшой вычислительной системы на основе микроЭВМ показана на рис. 1. Такая микроЭВМ содержит все 5 основных блоков цифровой машины: устройство ввода информации, управляющее устройство (УУ), арифметико-логическое устройство (АЛУ) (входящие в состав микропроцессора), запоминающие устройства (ЗУ) и устройство вывода информации.

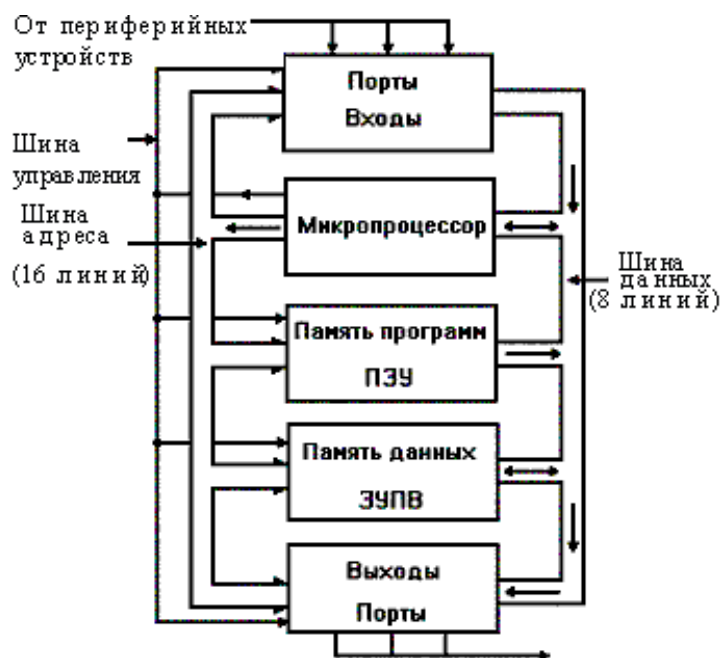


Рисунок 1 - Архитектура типового микропроцессора.

Микропроцессор координирует работу всех устройств цифровой системы с помощью шины управления (ШУ). Помимо ШУ имеется 16-разрядная адресная шина (ША), которая служит для выбора определенной ячейки памяти, порта ввода или порта вывода. По 8-разрядной информационной шине или шине данных (ШД) осуществляется двунаправленная пересылка данных к микропроцессору и от микропроцессора. Важно отметить, что МП может посылать информацию в память микроЭВМ или к одному из портов вывода, а также получать информацию из памяти или от одного из портов ввода.

Постоянное запоминающее устройство (ПЗУ) в микроЭВМ содержит некоторую программу (на практике программу инициализации ЭВМ). Программы могут быть загружены в запоминающее устройство с произвольной выборкой (ЗУПВ) и из внешнего запоминающего устройства (ВЗУ). Это программы пользователя.

2 Основные блоки процессора

Главным элементом или «мозгом» компьютера является микропроцессор, который выполняет обработку всех команд и данных и имеет 32-разрядную шину адреса и 16-ти разрядную шину данных.

Обычно процесс выполнения команд в МП включает в себя следующие этапы:

1. Извлечение кода команды (операндов команды, если это требует код команды) из памяти;
2. Выполнение команды;
3. Запись результата (если этого требует команда).

В МП два отдельных процессорных блока (рис. 2): операционное устройство (ОУ) и шинный интерфейс (ШИ). Работа ОУ заключается в декодировании и выполнении команд, в то время как ШИ подготавливает команды для выполнения, осуществляет связь с внешними устройствами, обеспечивает выборку команд и данных из памяти, формирует очередь команд. Операционное устройство содержит арифметико-логическое устройство (АЛУ) с регистром состояния или регистром флагов, устройство управления и регистры общего назначения. Эти устройства обеспечивают выполнение команд, арифметические вычисления и логические операции.

Три элемента шинного интерфейса (ШИ): устройство управления шиной, очередь команд и сегментные регистры осуществляют три важные функции: во-первых, ШИ управляет передачей данных на ОУ, память и внешнее устройство ввода-вывода; во-вторых, четыре сегментных регистра управляют объемом памяти до 1 Мбайта. Третья функция ШИ это выборка команд. Это позволяет ШИ выдавать в ОУ сохранённые команды по мере необходимости без загрузки внешних магистралей МП. Блок ШИ организует получение нового кода команды, как только 2 байта в очереди команд будут переданы в ОУ.

Пока ОУ занята выполнением первой команды, ШИ выбирает из очереди команд код следующей команды и передает его в операционное устройство последовательно так, как они записаны в программе. Если ОУ выполняет команду передачи управления в другое место программы, то ШИ очищает очередь команд, получает код команды из нового адреса, передаёт его в ОУ и начинает заполнять очередь заново.

3 Работа процессора. Организация обрабатываемой части микропроцессора

На рис. 2 а показано функциональное разбиение структуры процессора при создании трехкристального микропроцессора (пунктирные линии), содержащего БИС операционного (ОП), БИС управляющего (УП) и БИС интерфейсного (ИП) процессоров.

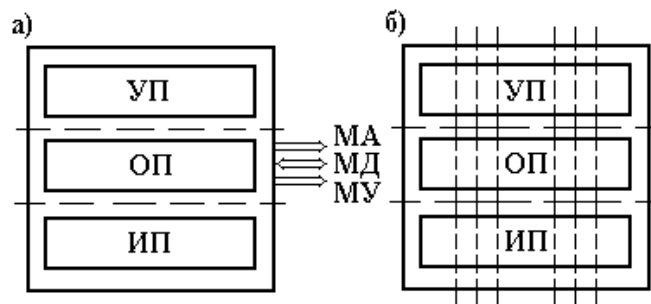


Рисунок 2 - Функциональная структура процессора (а) и ее разбиение для реализации процессора в виде комплекта секционных БИС

Операционный процессор служит для обработки данных, управляющий процессор выполняет функции выборки, декодирования и вычисления адресов операндов и также генерирует последовательности микрокоманд. Автономность работы и большое быстродействие БИС УП позволяет выбирать команды из памяти с большей скоростью, чем скорость их исполнения БИС ОП. При этом в УП образуется очередь еще не исполненных команд, а также заранее подготавливаются те данные, которые потребуются ОП в следующих циклах работы. Такая опережающая выборка команд экономит время ОП на ожидание операндов, необходимых для выполнения команд программ. Интерфейсный процессор позволяет подключить память и периферийные средства к микропроцессору; он, по существу, является сложным контроллером для устройств ввода/вывода информации. БИС ИП выполняет также функции канала прямого доступа к памяти.

Выбираемые из памяти команды распознаются и выполняются каждой частью микропроцессора автономно и поэтому может быть обеспечен режим одновременной работы всех БИС МП, т.е. конвейерный поточный режим исполнения последовательности команд программы (выполнение последовательности с небольшим временным сдвигом). Такой режим работы значительно повышает производительность микропроцессора.

Многокристальные секционные микропроцессоры получают в том случае, когда в виде БИС реализуются части (секции) логической структуры процессора при функциональном разбиении ее вертикальными плоскостями (рис. 1,б). Для построения многоразрядных микропроцессоров при параллельном включении секций БИС в них добавляются средства "стыковки".

Для создания высокопроизводительных многоразрядных микропроцессоров требуется столь много аппаратных средств, не реализуемых в доступных БИС, что может возникнуть необходимость еще и в функциональном разбиении структуры микропроцессора горизонтальными плоскостями. В результате рассмотренного функционального разделения структуры микропроцессора на функционально и конструктивно законченные части создаются условия реализации каждой из них в виде БИС. Все они образуют комплект секционных БИС МП.

Таким образом, микропроцессорная секция это БИС, предназначенная для обработки нескольких разрядов данных или выполнения определенных

управляющих операций. Секционность БИС МП определяет возможность "наращивания" разрядности обрабатываемых данных или усложнения устройств управления микропроцессора при "параллельном" включении большего числа БИС.

Однокристалльные и трехкристалльные БИС МП, как правило, изготавливают на основе микроэлектронных технологий униполярных полупроводниковых приборов, а многокристалльные секционные БИС МП на основе технологии биполярных полупроводниковых приборов. Использование многокристалльных микропроцессорных высокоскоростных биполярных БИС, имеющих функциональную законченность при малой физической разрядности обрабатываемых данных и монтируемых в корпус с большим числом выводов, позволяет организовать разветвление связи в процессоре, а также осуществить конвейерные принципы обработки информации для повышения его производительности.

Коды операции команд программы, воспринимаемые управляющей частью микропроцессора, расшифрованные и преобразованные в ней, дают информацию о том, какие операции надо выполнить, где в памяти расположены данные, куда надо направить результат и где расположена следующая за выполняемой команда.

Управляющее устройство имеет достаточно средств для того, чтобы после восприятия и интерпретации информации, получаемой в команде, обеспечить переключение (срабатывание) всех требуемых функциональных частей машины, а также для того, чтобы подвести к ним данные и воспринять полученные результаты. Именно срабатывание, т. е. изменение состояния двоичных логических элементов на противоположное, позволяет посредством коммутации вентиляей выполнять элементарные логические и арифметические действия, а также передавать требуемые операнды в функциональные части микроЭВМ.

Устройство управления в строгой последовательности в рамках тактовых и цикловых временных интервалов работы микропроцессора (такт - минимальный рабочий интервал, в течение которого совершается одно элементарное действие; цикл - интервал времени, в течение которого выполняется одна машинная операция) осуществляет: выборку команды; интерпретацию ее с целью анализа формата, служебных признаков и вычисления адреса операнда (операндов); установление номенклатуры и временной последовательности всех функциональных управляющих сигналов; генерацию управляющих импульсов и передачу их на управляющие шины функциональных частей микроЭВМ и вентили между ними; анализ результата операции и изменение своего состояния так, чтобы определить месторасположение (адрес) следующей команды.

В микропроцессорах используют два метода выработки совокупности функциональных управляющих сигналов: программный и микропрограммный.

Выполнение операций в машине сводится к элементарным преобразованиям информации (передача информации между узлами в блоках,

сдвиг информации в узлах, логические поразрядные операции, проверка условий и т.д.) в логических элементах, узлах и блоках под воздействием функциональных управляющих сигналов блоков (устройств) управления. Элементарные преобразования, неразложимые на более простые, выполняются в течение одного такта сигналов синхронизации и называются микрооперациями.

В аппаратных (схемных) устройствах управления каждой операции соответствует свой набор логических схем, вырабатывающих определенные функциональные сигналы для выполнения микроопераций в определенные моменты времени. При этом способе построения устройства управления реализация микроопераций достигается за счет однажды соединенных между собой логических схем, поэтому ЭВМ с аппаратным устройством управления называют ЭВМ с жесткой логикой управления. Это понятие относится к фиксации системы команд в структуре связей ЭВМ и означает практическую невозможность каких-либо изменений в системе команд ЭВМ после ее изготовления.

При микропрограммной реализации устройства управления в состав последнего вводится ЗУ, каждый разряд выходного кода которого определяет появление определенного функционального сигнала управления. Поэтому каждой микрооперации ставится в соответствие свой информационный код - микрокоманда. Набор микрокоманд и последовательность их реализации обеспечивают выполнение любой сложной операции. Набор микроопераций называют микропрограммами. Способ управления операциями путем последовательного считывания и интерпретации микрокоманд из ЗУ (наиболее часто в виде микропрограммного ЗУ используют быстродействующие программируемые логические матрицы), а также использования кодов микрокоманд для генерации функциональных управляющих сигналов называют микропрограммным, а микроЭВМ с таким способом управления - микропрограммными или с хранимой (гибкой) логикой управления.

К микропрограммам предъявляют требования функциональной полноты и минимальности. Первое требование необходимо для обеспечения возможности разработки микропрограмм любых машинных операций, а второе связано с желанием уменьшить объем используемого оборудования. Учет фактора быстродействия ведет к расширению микропрограмм, поскольку усложнение последних позволяет сократить время выполнения команд программы.

Преобразование информации выполняется в универсальном арифметико-логическом блоке микропроцессора. Он обычно строится на основе комбинационных логических схем.

Для ускорения выполнения определенных операций вводятся дополнительно специальные операционные узлы (например, циклические сдвигатели). Кроме того, в состав микропроцессорного комплекта (МПК) БИС вводятся специализированные оперативные блоки арифметических расширителей.

Операционные возможности микропроцессора можно расширить за счет увеличения числа регистров. Если в регистровом буфере закрепление функций регистров отсутствует, то их можно использовать как для хранения данных, так и для хранения адресов. Подобные регистры микропроцессора называются регистрами общего назначения (РОН). По мере развития технологии реально осуществлено изготовление в микропроцессоре 16, 32 и более регистров.

В целом же, принцип микропрограммного управления (ПМУ) включает следующие позиции:

1) любая операция, реализуемая устройством, является последовательностью элементарных действий - микроопераций;

2) для управления порядком следования микроопераций используются логические условия;

3) процесс выполнения операций в устройстве описывается в форме алгоритма, представляемого в терминах микроопераций и логических условий, называемого микропрограммой;

4) микропрограмма используется как форма представления функции устройства, на основе которой определяются структура и порядок функционирования устройства во времени.

ПМУ обеспечивает гибкость микропроцессорной системы и позволяет осуществлять проблемную ориентацию микро- и миниЭВМ.

4 RISC – процессоры с сокращенной системой команд и CISC (Complete Instruction Set Computer) процессоры с полной системой команд.

RISC (*Reduced Instruction Set Computer*) – архитектура процессора с сокращённым набором инструкций. Начало исследований данной области положено компанией IBM (в исследовательском центре IBM, имени *Томаса Джона Уотсона*) в 1975 году.

Была создана для устранения недостатков CISC архитектуры, но не получила популярности в то время из-за унификации стандарта Intel x86 и всех программ выпущенных в то время под CISC процессоры (точнее нежелания их переписывать заново). «Сокращённый набор команд» вовсе не означает, что процессор имеет малое количество инструкций. Это значит лишь то, что инструкции разделены на действия, результаты которых могут быть вычислены за определённый период времени (обычно один такт).

Начиная с Intel 486DX все x86 процессоры имеют внутреннее ядро RISC, остался только преобразователь и дополнительные конвейеры, который на входе преобразует CISC инструкции в RISC, а на выходе обратно в CISC. Это необходимо из-за особенностей архитектуры x86, но иногда тормозит работу процессора и увеличивается количество транзисторов, площадь и тепловыделение в сравнении с полноценными RISC процессорами.

Появление полноценной RISC архитектуры на процессорах, позволило упростить конструкцию вычислительных ядер; уменьшить стоимость, площадь и при этом увеличить количество регистров общего назначения; унифицировать команды для вычислительных ядер и сравнять время выполнения всех команд,

что также позволило воплотить в жизнь конвейерную обработку инструкций (реализация сложных инструкций из множества результатов более простых).

Вычислительным ядрам больше не нужно было обращаться к более медленной оперативной памяти для занесения и считывания результатов. Эти цели теперь выполняют регистры общего назначения, а к оперативной памяти обращение идёт только в процессе чтения начальных данных и вывода результатов вычислений. Поддерживается маршрут «регистр-регистр».

*в сравнении с CISC – архитектурой

RISC архитектура сейчас.

Основной проблемой по реализации RISC архитектуры являлась недостаточная поддержка со стороны софта и программного обеспечения. Но с появлением поддержки UNIX/Linux подобных систем, эта проблема практически решилась.

Самыми известными и успешными представителями архитектуры RISC являются ARM, разработанная ARM Holdings. Процессоры с данной архитектурой, применяемые в абсолютном большинстве мобильных устройств и даже серверных системах, благодаря очень низкому энергопотреблению и тепловыделению.



Из наиболее известных **RISC** архитектур можно также выделить разработки компании **IBM** (*PowerPC, Power*), компанией **Sun Microsystems** (*SPARC- архитектура*) и компанией **Atmel** (*AVR – архитектура*).

На данный момент, **RISC** – архитектура является одной *самых распространённых в мире*, имея более **40%** мирового **рынка**. Данный результат в основном благодаря **ARM** архитектуре и то, что в современных мобильных устройствах используются именно процессоры **ARM** (в абсолютном большинстве).

Микропроцессор CISC использует набор машинных инструкций, полностью соответствующий набору команд языка ассемблера. Вычисления разного типа в нем могут выполняться различными командами, даже если они приводят к одному результату (например, умножение на два и сдвиг на один разряд влево). Такая архитектура обеспечивает разнообразные и мощные

способы выполнения вычислительных операций на уровне машинных команд, но для выполнения каждой команды обычно требуется большое число тактов процессора.

Для CISC-процессоров характерно:

- сравнительно небольшое число регистров общего назначения;
- большое количество машинных команд, некоторые из которых нагружены семантически аналогично операторам высокоуровневых языков программирования и выполняются за много тактов;
- большое количество методов адресации;
- большое количество форматов команд различной разрядности;
- преобладание двухадресного формата команд; наличие команд обработки типа регистр-память.

Организация первых моделей процессоров - i8086/8088 - была направлена, в частности, на сокращение объёма программ, критичного для систем того времени, отличавшихся малой оперативной памятью. Расширение спектра операций, реализуемых системой команд, позволило уменьшить размер программ, а также трудоёмкость их написания и отладки. Однако увеличение числа команд повысило трудоёмкость разработки их топологических и микропрограммных реализаций. Последнее проявилось в удлинении сроков разработки CISC-процессоров, а также в проявлении различных ошибок в их работе. Кроме того, нерегулярность потока команд ограничила развитие топологии временным параллелизмом обработки инструкций на конвейере “выборка команды - дешифрация команды- выборка данных- вычисление- запись результата”.

Эти недостатки обусловили необходимость разработки альтернативной архитектуры, нацеленной, прежде всего, на снижение нерегулярности потока команд уменьшением их общего количества. Это было реализовано в RISC-процессорах, название которых означает “чипы с сокращённой системой команд” (Reduced Instruction Set Computer).

Контрольные вопросы:

1. Какова обобщенная структура арифметических процессоров?
2. Как взаимодействуют компоненты процессора в процессе обработки информации?
3. Назовите основные блоки процессора.
4. Что означает магистрально-модульный принцип построения ЭВМ?
5. Что включает в себя принцип микропрограммного управления?
6. Материнские платы.
7. Виды архитектур процессоров, их особенности.

Тема 8. Центральное устройство управления (ЦУУ)

1. Центральное устройство управления (ЦУУ). Основные функции ЦУУ.
2. Выполнение программ в процессоре.
3. Управление памятью и внешними устройствами.
4. Средства микропрограммной работы процессора.
5. Принципы организации прерываний. Многоуровневые системы прерываний. Приоритеты.
6. Средства защиты памяти при организации мультипрограммного режима работы.

1 Центральное устройство управления (ЦУУ). Основные функции ЦУУ

Устройство управления (УУ) обеспечивает автоматическое управление процессом вычислений, координацию работы всех устройств ЦВМ посредством выработки соответствующих синхронизирующих и управляющих сигналов.

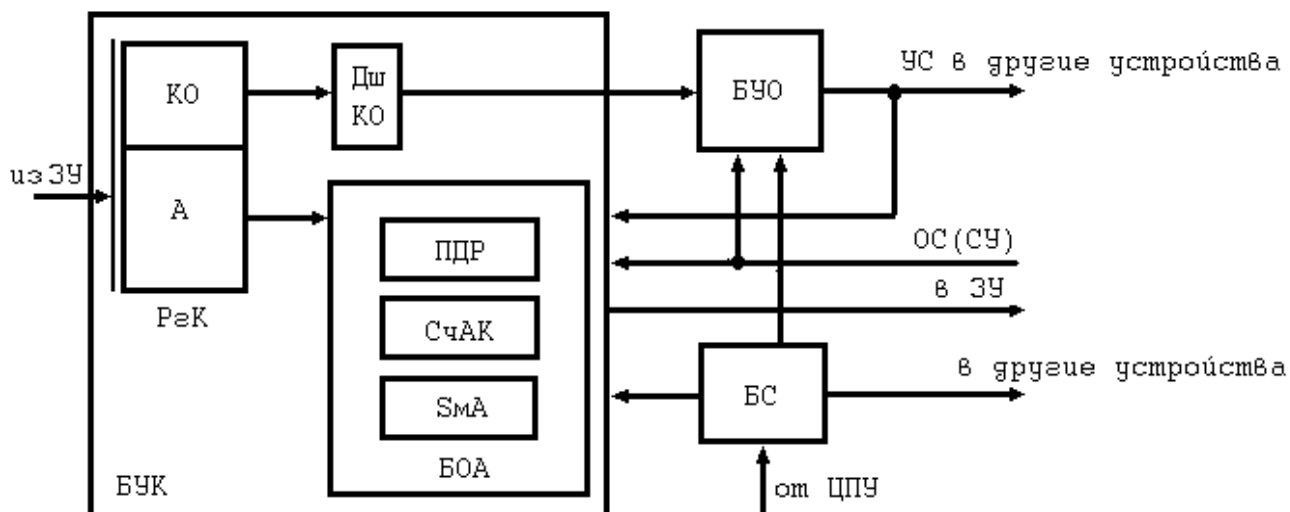


Рисунок 1 - Обобщенная структурная схема УУ

БУК – блок управления командой – служит для выборки, хранения и дешифрации кода очередной команды и формирует исполнительный адрес следующей команды. Содержит:

РгК – регистр команд. Принимает на выполнение очередную команду из ЗУ, которая состоит из кода операции (КО) и адресной части (А).

Дш КО – дешифратор кода операций;

БОА – блок обработки адресов. Выполняет формирование адреса очередной команды и включает:

ПДР – программно-доступные регистры;

Сч АК – счетчик адреса команд;

СМА – сумматор адресов.

БУО – блок управления операцией. Вырабатывает управляющие сигналы с учетом:

КО – кода операции;

ОС – оповещающих сигналов;

УС – сигналами условий;

Сигналы ОС и УС вырабатываются другими устройствами ЭВМ.

БС – блок синхронизации вырабатывает тактовые и синхронизирующие сигналы;

В зависимости от способа организации БУО УУ подразделяются на:

- УУ со схемным формированием УС (УУ с «жесткой» логикой);
- УУ с программируемой логикой (микропрограммные УУ).

В БУО с «жесткой» логикой последовательность управляющих сигналов вырабатывается посредством логических схем. Для выполнения команд «разной» (по времени) длины используются дополнительные счетчики тактов.

Такие БУО строятся на основе принципа интерпретации микропрограмм для выполнения операций. Для чего на языке микроопераций описывается микропрограмма выполнения какой-либо команды из системы машинных команд ЭВМ. На основе микропрограмм выполнения машинных операций строится управляющий автомат (Мили или Мура).

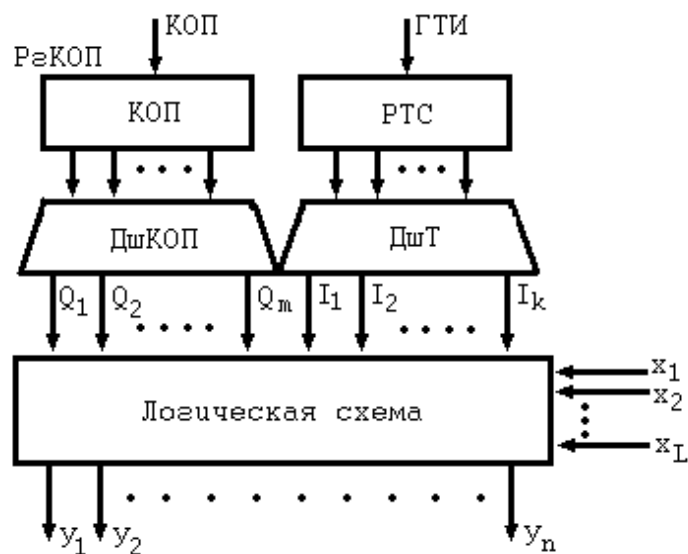


Рисунок 2 - Обобщенная структурная схема УУ с «жесткой» логикой

ГТИ – генератор тактовых импульсов;

РТС – распределитель тактовых сигналов;

Q_1, Q_2, \dots, Q_m – m-команд системы машинных команд ЭВМ;

I_1, I_2, \dots, I_k – количество числа тактов, необходимых для выполнения самой длинной операции;

x_1, x_2, \dots, x_L – сигналы логических условий;

y_1, y_2, \dots, y_n – управляющие сигналы.

В БУО микропрограммного типа каждой микропрограмме соответствует свое управляющее слово. Такое слово определяет порядок функционирования устройства в течении одного такта. Микропрограмма выполнения каждой машинной команды из системы команд состоит из последовательности **микрокоманд** и хранится в памяти микропрограмм «постоянного» типа. Таким образом, можно менять набор микрокоманд непосредственно в процессе вычислений путем перепрограммирования МП.

БУО микропрограммного типа имеют следующие преимущества:

- большая гибкость;
- регулярная структура (длина микрокоманды – постоянна и не зависит от «длины» команды);
- возможность построения эффективной схемы микродиагностики.

2 Выполнение программ в процессоре

Рассмотрим упрощенную схему выполнения процессором машинных команд программы. Эта схема на логическом уровне адекватна реальной работе процессора. Отличие состоит в том, что в реальном процессоре с помощью различных технических приемов организуется параллельное, то есть одновременное выполнение нескольких этапов одной и той же команды или нескольких последовательных команд программы, за счет чего добиваются значительного ускорения работы процессора.

Перед запуском программы на выполнение операционная система автоматически записывает в регистр адрес сегмента кода, который, как указывалось ранее, содержит код выполняющейся программы. Допустим, что операционная система занесла в регистр код. Первые 25610 байтов сегмента кода операционная система всегда занимает вспомогательной информацией. Этот участок памяти принято называть префиксом программного сегмента, часто встречается также его обозначение PSP (от program segment prefix). Поэтому первый байт программы всегда имеет внутрисегментное смещение. Как известно, для задания адресов в сегменте кода используется пара регистров. Следовательно, операционная система перед началом выполнения программы должна записать в регистр код 010016.

Пусть в оперативной памяти в поле с адресом находятся коды, которые мы будем рассматривать как машинную программу, а регистры содержат выбранные ранее коды 100016 и 010016 соответственно. Из всех остальных регистров процессора в данном примере нас интересует только содержимое регистра di, которое мы примем равным, например, 003A16. Работу процессора при выполнении машинной команды удобно разбить на несколько этапов.

3 Управление памятью и внешними устройствами

В современных операционных системах, таких как Windows, приложения и многие системные процессы *всегда* ссылаются на память с помощью адресов виртуальной памяти. Эти адреса автоматически преобразуются оборудованием

в реальные адреса (ОЗУ). Только базовые части ядра операционной системы пропускают это преобразование и напрямую используют реальные адреса.

Виртуальная память применяется всегда, даже если объем памяти, требуемый всем запущенным процессам, не превышает объем ОЗУ, установленный в системе.

Всем процессам (например, исполняемым файлам приложений), работающим в 32-разрядных версиях Windows, назначаются виртуальные адреса (*виртуальное адресное пространство*) в диапазоне от 0 до 4 294 967 295 ($2^{32}-1 = 4 \text{ ГБ}$) независимо от того, сколько фактической оперативной памяти доступно на компьютере.

В конфигурации Windows по умолчанию 2 гигабайта (ГБ) этого виртуального адресного пространства выделены каждому процессу для частного использования, а другие 2 ГБ совместно применяются всеми процессами и операционной системой. Обычно приложения (например, Блокнот, Word, Excel и Acrobat Reader) используют только часть из 2 ГБ частного адресного пространства. Операционная система назначает рамки страниц ОЗУ только используемым страницам виртуальной памяти.

Расширение физических адресов (PAE) — это возможность 32-разрядной архитектуры Intel, которая расширяет адреса физической памяти (ОЗУ) до 36 разрядов. PAE не меняет размер виртуального адресного пространства (он остается равным 4 ГБ), а только объем фактической оперативной памяти, которая доступна процессору.

Преобразование 32-разрядных адресов виртуальной памяти, используемое в коде, выполняемом в процессе, в 36-разрядный адрес ОЗУ осуществляется автоматически и прозрачно оборудованием компьютера в соответствии с *таблицами преобразования*, которые создаются операционной системой. Любая страница виртуальной памяти (32-разрядный адрес) может быть связана с любой страницей физической памяти (36-разрядный адрес).

Оперативная память — это ограниченный ресурс, а виртуальная память для практических целей не ограничена. Может существовать множество процессов, каждому из которых выделено собственное частное виртуальное адресное пространство объемом 2 ГБ. Если объем памяти, используемый всеми текущими процессами, превышает объем ОЗУ, операционная система перемещает страницы (размером 4 КБ) одного или нескольких виртуальных адресных пространств на жесткий диск компьютера. Это освобождает ОЗУ для других целей. В системах Windows такие страницы хранятся в одном или нескольких файлах Pagefile.sys в корневом каталоге раздела. В каждом разделе может быть один такой файл. Размещение и размер файла подкачки задаются в **свойствах системы** (последовательно выберите **Дополнительно**, **Производительность** и нажмите кнопку **Параметры**).

Пользователи часто спрашивают, какой максимальный размер следует задать для файла подкачки. Единственного правильного ответа на этот вопрос нет, так как размер зависит от объема ОЗУ и требуемого объема виртуальной памяти. Если другие сведения недоступны, типичный рекомендуемый размер файла подкачки равен объему ОЗУ $\times 1,5$. На серверных системах обычно

требуется достаточный объем ОЗУ, чтобы избежать дефицита оперативной памяти, поэтому файл подкачки, по сути, не используется. В этих системах большой файл подкачки не несет полезной нагрузки. С другой стороны, если места на диске достаточно, крупный файл подкачки (например, в 1,5 раза больше объема ОЗУ) не вызовет проблем и позволит не волноваться о том, какой размер следует использовать.

Вычислительные машины, помимо процессоров и основной памяти (образующих ее ядро), содержат многочисленные периферийные устройства (ПУ): ВЗУ и УВВ.

Передача информации с периферийного устройства в ЭВМ называется *операцией ввода*, а передача из ЭВМ в ПУ — *операцией вывода*.

Производительность и эффективность ЭВМ определяются не только возможностями ее процессора и характеристиками ОП, но и составом ПУ, их техническими данными и способами организации их совместной работы с ЭВМ.

При разработке систем ввода-вывода ЭВМ особое внимание обращается на решение следующих проблем:

- должна быть обеспечена возможность реализации машин с переменным составом оборудования (машин с переменной конфигурацией), в первую очередь с различным набором периферийных устройств, с тем чтобы пользователь мог выбирать состав оборудования (конфигурацию) машины в соответствии с ее назначением, легко дополнять машину новыми устройствами;
- для эффективного и высокопроизводительного использования оборудования в ЭВМ должны реализовываться одновременная работа процессора над программой и выполнение периферийными устройствами процедур ввода-вывода;
- необходимо упростить для пользователя и стандартизировать программирование операций ввода-вывода, обеспечить независимость программирования ввода-вывода от особенностей того или иного периферийного устройства;
- необходимо обеспечить автоматическое распознавание и реакцию ядра ЭВМ на многообразие ситуаций, возникающих в ПУ (готовность устройства, отсутствие носителя, различные нарушения нормальной работы и др.).

Первый шаг в совершенствовании структуры ЭВМ с целью повышения производительности был сделан при разработке второго поколения ЭВМ, когда впервые была обеспечена автономность внешних устройств. Сначала она была необходима для согласования скорости работы устройств ЭВМ.

Однако автономность потребовала проведения проверок исправности устройств при каждом обращении к ним. Появилась определенная технология обращения к автономным внешним устройствам — интерфейс.

Когда эту технологию унифицировали, появилось понятие стандартного интерфейса.

Стандартизация интерфейсов ввода-вывода привела к возможности гибко изменять конфигурацию вычислительных машин (количество и состав внешних устройств, расширять комплект ЭВМ за счет подключения новых устройств).

Затем появилась концепция виртуальных устройств, позволяющая совмещать различные типы ЭВМ, операционные системы. Совместимость распространилась и на работу машин разной конфигурации (можно использовать соответствующее программное обеспечение и при физическом отсутствии необходимых устройств).

Изменилась и технология работы — при отсутствии печатающих устройств файлы направлялись в виртуальное устройство, где и накапливались, а реально распечатывались на другой машине.

Дальнейшее совершенствование интерфейсов потребовалось при решении специфических задач: новые внешние устройства (сканеры) позволяли вводить текст в графическом виде, а обрабатывать его надо было в символьном виде. Поэтому возникла необходимость в системах распознавания, идентификации, преобразования из графического формата в символьный.

При анализе снимков из космоса появилась необходимость автоматической классификации наблюдаемых объектов, самообучения распознающей программы, анализа сцен. Все это стимулировало развитие специального математического аппарата в интерфейсах.

Необходимость вывода информации различными шрифтами привела к изменению технологии вывода, связанной с применением TrueType-шрифтов, масштабированием матричных и векторных изображений, преобразованием векторных символов в матричные.

Во внешние устройства стали встраивать свою основную память, объем которой должен обеспечивать вывод целой страницы (например, текста) при плотности 600 точек/дюйм — этот объем превышает иногда объем основной памяти ЭВМ.

Необходимость пересчета при преобразовании векторных шрифтов в матричные привела к необходимости включить в состав устройств отображения информации специальные графические процессоры матричной архитектуры, в качестве которых часто используются транспьютеры.

При создании автоматизированных рабочих мест появилась необходимость подключения к ЭВМ аудиовизуальной аппаратуры — телекамер, видеомэгнитофонов, фотокамер, аудиоустройств. На этой основе образовался специальный аппаратный комплекс, называемый *анимационной линейкой*. Из-за технической несовместимости необходимой для таких комплексов аппаратуры потребовались программные и аппаратные преобразователи сигналов, специальные средства сопряжения.

При разработке пятого поколения ЭВМ была заложена интеллектуализация общения: речевой ввод и вывод; графический ввод; ввод текстовой информации без клавиатуры; естественно-языковое общение; общение на разных естественных языках.

Все это и явилось основой для совершенствования систем, обеспечивающих связь ЭВМ с периферийными устройствами.

В общем случае для организации и проведения обмена данными между двумя устройствами требуются специальные средства:

- специальные управляющие сигналы и их последовательности;

- устройства сопряжения;
- линии связи;
- программы, реализующие обмен.

Весь этот комплекс линий и шин, сигналов, электронных схем, алгоритмов и программ, предназначенный для осуществления обмена информацией, называется **интерфейсом**.

В зависимости от типа соединяемых устройств различаются:

- внутренний интерфейс ЭВМ (например, интерфейс системной шины, НМД), предназначенный для сопряжения элементов внутри системного блока ПЭВМ;
- интерфейс ввода-вывода — для сопряжения различных устройств с системным блоком (клавиатурой, принтером, сканером, мышью, дисплеем и др.);
- интерфейсы межмашинного обмена (для обмена между разными машинами) — для сопряжения различных ЭВМ (например, при образовании вычислительных сетей);
- интерфейсы «человек — машина» — для обмена информацией между человеком и ЭВМ.

Для каждого интерфейса характерно наличие специального аппаратного комплекса (рис. 3).

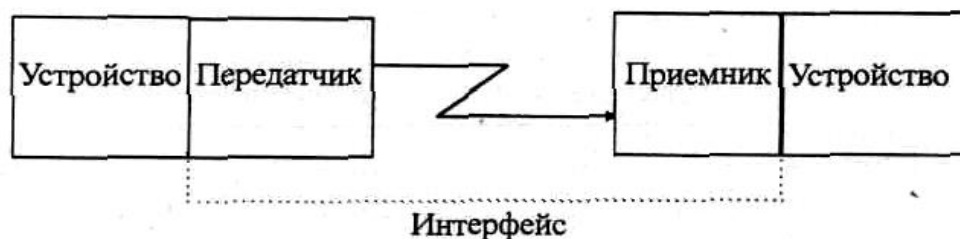


Рисунок 3

Характер использования этого аппаратного комплекса определяется технологией обмена, установленным церемониалом, идеологией данного интерфейса, протоколом общения.

Особое место занимает человеко-машинный интерфейс, реализующий специфические методы организации общения. Например, церемониал человеко-машинного общения предусматривает «дружелюбность пользователю», которая может обеспечиваться техническими средствами, программами и технологией общения.

Если интерфейс обеспечивает обмен одновременно всеми разрядами передаваемой информационной единицы (чаще всего — байта или машинного слова), он называется **параллельным интерфейсом**.

Внутренний интерфейс ЭВМ всегда делается параллельным или последовательно-параллельным (если одновременно передается не вся информационная единица, а ее часть, содержащая несколько двоичных

разрядов: например, i8088 обеспечивает побайтовый, а 80386SX — двухбайтовый обмен через системную магистраль, тогда как основная информационная единица — машинное слово — имеет длину: у 8086 — 2 байта, а у 80386 — 4 байта).

Интерфейсы межмашинного обмена обычно последовательные, т.е. в них обмен информацией производится по одному биту, последовательно.

Для параллельного и последовательно-параллельного интерфейса необходимо, чтобы участники общения были связаны многожильным интерфейсным кабелем (количество жил не меньше числа одновременно передаваемых разрядов — бит). В последовательных интерфейсах участники общения связываются друг с другом одно-двух-проводной линией связи, световодом, коаксиальным кабелем, радиоканалом.

В зависимости от используемых при обмене программно-технических средств интерфейсы ввода-вывода делятся на два уровня: физический и логический (рис. 4).

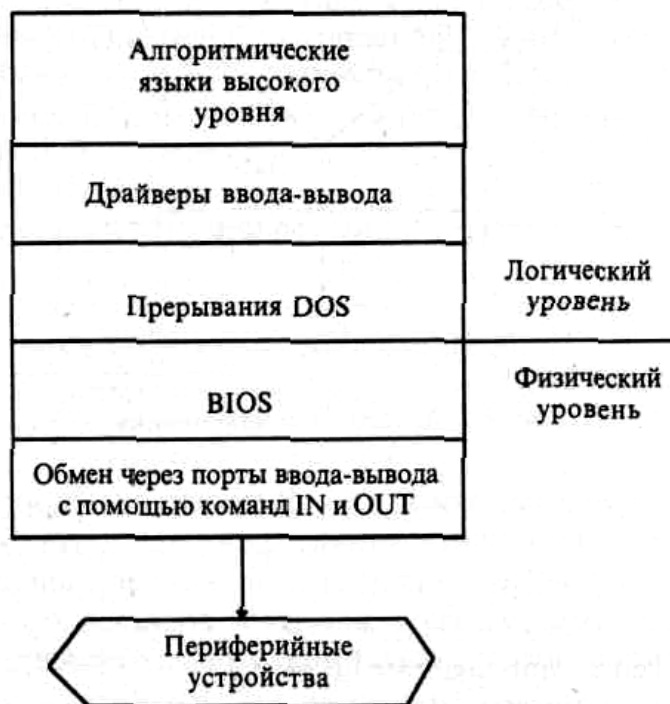


Рисунок 4 - Логический и физический уровни интерфейсов ввода-вывода.

В зависимости от степени участия центрального процессора в обмене данными в интерфейсах может использоваться три способа управления обменом:

- режим сканирования (так называемый «асинхронный» обмен);
- синхронный обмен;
- прямой доступ к памяти.

Для внутреннего интерфейса ЭВМ режим сканирования предусматривает опрос центральным процессором периферийного устройства (ПФУ): готово ли

оно к обмену, и если нет, то продолжается опрос периферийного устройства (рис. 5).

Операция пересылки данных логически слишком проста, чтобы эффективно загружать сложную быстродействующую аппаратуру процессора, в результате чего в режиме сканирования снижается производительность вычислительной машины.

Вместе с тем при пересылке блока данных процессору приходится для каждой единицы передаваемых данных (байт, слово) выполнять довольно много команд (чтобы обеспечить буферизацию данных, преобразование форматов, подсчет количества переданных данных, формирование адресов в памяти и т.п.). В результате скорость передачи данных при пересылке блока данных даже через высокопроизводительный процессор может оказаться неприемлемой для систем управления, работающих в реальном масштабе времени.



Рисунок 5 - Алгоритм сканирования

Режим сканирования упрощает подготовку к обмену, но имеет ряд недостатков:

- процессор постоянно задействован и не может выполнять другую работу;
- при большом быстродействии периферийного устройства процессор не успевает организовать обмен данными.

В синхронном режиме центральный процессор выполняет основную роль по организации обмена, но в отличие от режима сканирования не ждет готовности устройства, а осуществляет другую работу. Когда в нем возникает нужда, внешнее устройство с помощью соответствующего прерывания обращает на себя внимание центрального процессора.

Для быстрого ввода-вывода блоков данных и разгрузки процессора от управления операциями ввода-вывода используют прямой доступ к памяти (DMA — Direct Memory Access).

Прямым доступом к памяти называется *способ обмена данными*, обеспечивающий автономно от процессора установление связи и передачу данных между основной памятью и внешним устройством.

В режиме прямого доступа к памяти используется специализированное устройство — контроллер прямого доступа к памяти, который перед началом обмена программируется с помощью центрального процессора: в него передаются адреса основной памяти и количество передаваемых данных. Затем центральный процессор от контроллера прямого доступа к памяти отключается, разрешив ему работать, и до окончания обмена может выполнять другую работу. Об окончании обмена контроллер прямого доступа к памяти сообщает процессору. В этом случае участие центрального процессора косвенное. Обмен ведет контроллер прямого доступа к памяти. Прямой доступ к памяти (ПДП):

- освобождает процессор от управления операциями ввода-вывода;
- позволяет осуществлять параллельно во времени выполнение процессором программы с обменом данными между внешним устройством и основной памятью;
- производит обмен данными со скоростью, ограничиваемой только пропускной способностью основной памяти и внешним устройством.

ПДП разгружает процессор от обслуживания операций ввода-вывода, способствует увеличению общей производительности ЭВМ, дает возможность машине более приспособленно работать в системах реального времени.

При работе в режиме прямого доступа к памяти (ПДП) контроллер ПДП выполняет следующие функции:

- принимает запрос на ПДП от внешнего устройства;
- формирует запрос микропроцессору на захват шин системной магистрали;
- принимает сигнал, подтверждающий вход микропроцессора в состояние захвата (перехода в z-состояние, при котором процессор отключается от системной магистрали);
- формирует сигнал, сообщающий внешнему устройству о начале выполнения циклов ПДП;
- выдает на шину адреса системной магистрали адрес ячейки ОП, предназначенной для обмена;
- вырабатывает сигналы, обеспечивающие управление обменом данными;
- по окончании ПДП либо организует повторение цикла ПДП, либо прекращает режим ПДП, снимая запросы на него.

Циклы ПДП выполняются с последовательно расположенными ячейками памяти, поэтому контроллер ПДП имеет счетчик числа переданных байтов.

На рис. 6 приведена схема взаимодействия устройств микропроцессорной системы в режиме ПДП.

Контроллеры ПДП (КПДП) в IBM PC совместимы снизу вверх, т.е. программы, написанные с использованием КПДП ранних моделей ЭВМ, нормально работают и на более поздних моделях.

Каждый канал КПДП состоит из четырех 16-разрядных регистров (рис. 6): регистр текущего адреса (САР), регистр циклов ПДП (СWR), регистр хранения базового адреса (ВАР), регистр хранения базового числа циклов ПДП (WCR) и 6-разрядного регистра режима (MR).

Регистр текущего адреса хранит текущий адрес ячейки памяти при выполнении цикла ПДП. После выполнения каждого цикла ПДП содержимое этого регистра увеличивается или уменьшается на единицу. Оно может быть прочитано или загружено с помощью двух команд IN или OUT соответственно, каждая из которых работает только с одним байтом. В режиме автоинициализации содержимое регистра текущего адреса при обновлении принимает базовый адрес из регистра хранения базового адреса.

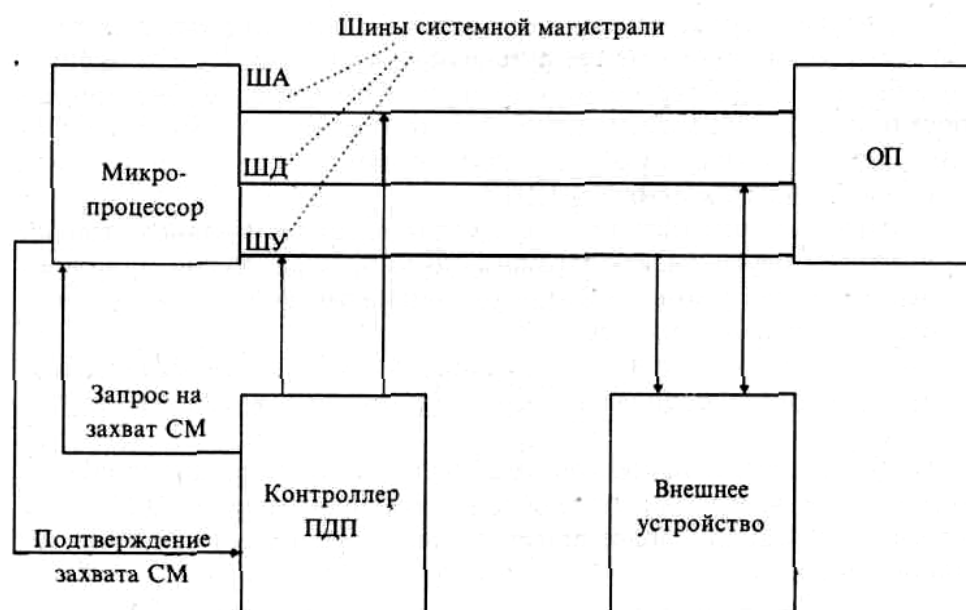


Рисунок 6 - Взаимодействие устройств в режиме ПДП.

Регистр циклов ПДП хранит число слов, предназначенных для передачи. При выполнении циклов регистр работает в режиме вычитающего счетчика. При переходе из нулевого состояния в FFFFH вырабатывается управляющий сигнал для блока управления контроллером. Регистр может быть прочитан двумя командами IN. В него можно осуществить запись двумя командами OUT из микропроцессора или в режиме инициализации — из регистра хранения базового числа циклов ПДП.

Регистры BAR и WCR предназначены для хранения констант — базового адреса и базового числа циклов. Они загружаются в режиме программирования КПДП одновременно с регистрами CAR и CWR. В процессе выполнения циклов ПДП их содержимое не изменяется. Прочитать состояние этих регистров невозможно.

Регистр режима определяет режим работы канала. Он содержит информацию о номере канала, типе цикла ПДП (чтение (ОП ← ВУ), запись (ОП → ВУ), проверка), необходимости автоинициализации, режиме изменения регистра текущего адреса (CAR) — увеличение или уменьшение и режиме работы канала — передача по запросу, одиночная передача, блочная передача, каскадирование (работа в составе каскада КПДП).

Регистр команд блока управления режимом определяет основные параметры работы канала. Загружается при программировании КПДП микропроцессором. Регистр условий хранит разрешение на ПДП каждому каналу (устанавливаемые программно) и запоминает факт перехода через 0 в регистре хранения базового числа циклов каждого канала.

Контроллер ПДП может работать в двух основных режимах: в режиме программирования и в режиме выполнения циклов ПДП.

В режиме программирования микропроцессор работает с КПДП, как с внешним устройством. После загрузки в КПДП управляющих слов контроллер переходит в пассивное состояние. В этом состоянии КПДП будет находиться до тех пор, пока не поступит запрос на ПДП от ВУ или от микропроцессора. Обнаружив запрос на ПДП, контроллер выставляет микропроцессору запрос на захват системной магистрали и ожидает от него подтверждения захвата (т.е. отключения МП от СМ, перехода его выходов в состояние высокого сопротивления, г-состояния). При получении сигнала подтверждения захвата, контроллер начинает выполнять циклы ПДП.

Системная магистраль (СМ) в режиме прямого доступа к памяти используется мультиплексно КПДП и микропроцессором, причем основное управление системной магистралью выполняет КПДП, а МП получает к ней доступ на очень короткие промежутки времени для обмена информацией с основной памятью.

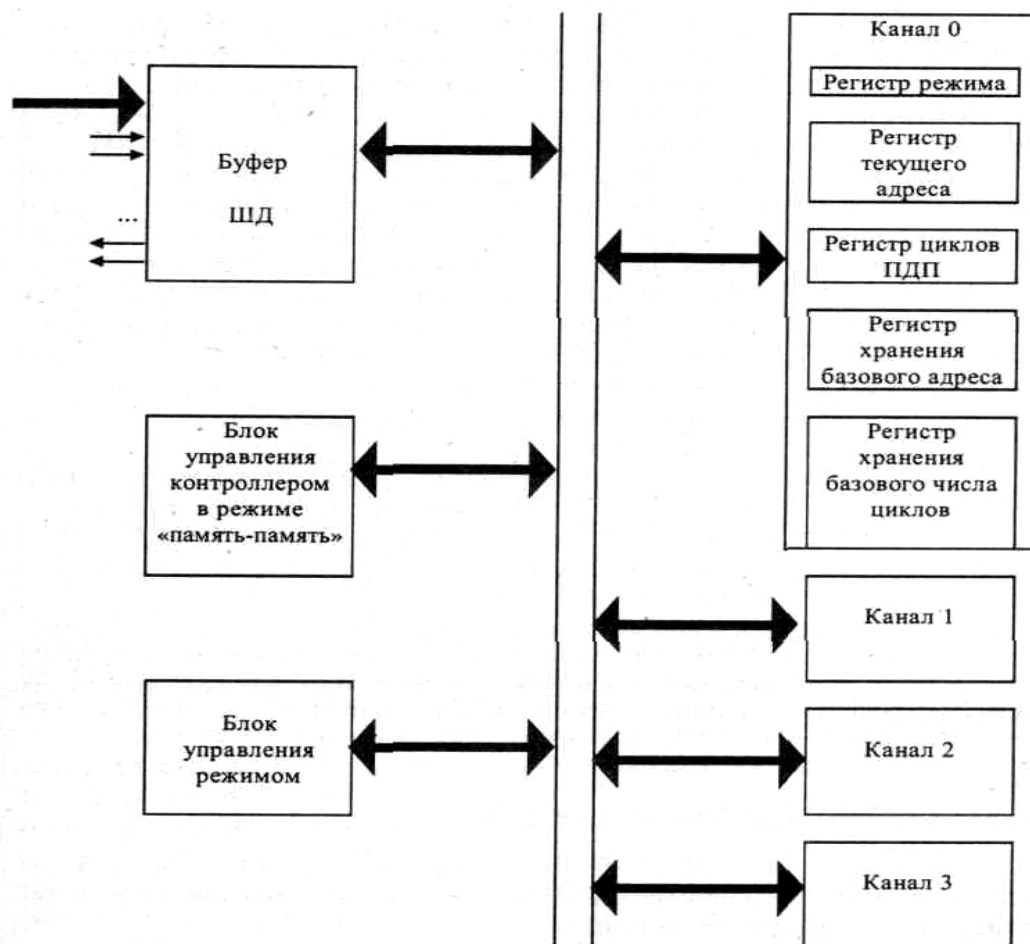


Рисунок 7 - Структурная схема контроллера ПДП.

Несмотря на большую самостоятельность ПДП, ведущее положение в МП-системе остается все-таки за микропроцессором, который, запустив обмен информацией в режиме ПДП, продолжает выполнять свою работу.

4 Средства микропрограммной работы процессора

Известны два подхода к построению логики формирования управляющих импульсов (УИ), необходимых для работы операционных устройств (ОУ) процессора. Один из них заключается в том, что каждой операции процессора соответствует набор логических (комбинационных) схем, формирующих управляющие импульсы (УИ), необходимые для работы ОУ. Если подать сигналы, соответствующие коду микрооперации (МО), а также сигналы условий выполнения этой МО на входы комбинационной схемы (КС), то требуемое действие будет выполнено с помощью определенных УИ на выходах КС в момент подачи тактового сигнала. Набор таких КС, оптимизированных для формирования УИ, управляющих работой ОУ при выполнении совокупности определенных МО, называется управляющим автоматом (УА). Такой принцип формирования УИ получил название **жесткой логики** и широко применяется во многих компьютерах. Преимуществом такой способа

формирования УИ является большое быстродействие при выполнении МО, т.к. большинство из них выполняются за один такт работы процессора. Недостатками являются большие аппаратные затраты и трудности при внесении изменений в работу ОУ.

Другой принцип организации управления ОУ, заключается в том, что каждой МО ставится в соответствие слово (или часть слова), называемое микрокомандой (МК) и хранимое в памяти внутри процессора подобно тому, как хранятся в памяти команды компьютера. Каждой команде процессора соответствует микропрограмма (МП), т. е. набор МК указывающих, какие УИ и в какой последовательности необходимо возбуждать для выполнения данной операции. Такой подход получил название **микропрограммирования или мягкой логики**. При микропрограммной реализации команд процессора логика управления ОУ реализуется не в виде электронной схемы, а в виде закодированной информации, находящейся в каком-то массиве регистров процессора или памяти. Как правило, микропрограммы хранятся в постоянном запоминающем устройстве (ПЗУ), находящемся внутри процессора. По сути при такой реализации команд процессора, внутри него используется УА, построенный по принципу фон-неймоновской вычислительной машины.

Основные положения, на которых основан принцип микропрограммного управления ОУ, заключаются в следующих пунктах :

1. Любая операция процессора рассматривается как сложное действие и разделяется на совокупность элементарных действий, называемых микрооперациями (МО). Выполнение каждой МО осуществляется специальной комбинационной схемой (КС) за один такт машинного времени;

2. Порядок выполнения МО задается алгоритмом операции и зависит от значений логических условий (ЛУ). ЛУ принимают значения истина или ложь в зависимости от результатов операции. ЛУ используются в качестве условий альтернативных переходов в алгоритмах операций;

3. Алгоритм, представленный, записанный в терминах МО и ЛУ, называется микропрограммой (МП). МП задает порядок выполнения МО и проверки ЛУ во времени.

4. Совокупность микропрограмм реализует любую команду процессора.

Преимуществом микропрограммного управления ОУ является, легкость модернизации процессора, которая может заключаться в изменении кода имеющихся МП или внесении новых, расширяющих систему команд процессора. К недостаткам можно отнести увеличение времени выполнения команд, т.к. появляются дополнительные операции, обеспечивающие выполнение микропрограмм. Данный недостаток, с развитием элементной базы стал несущественным, т.к. достигнуто значительное увеличение быстродействия элементов СБИС.

5 Принципы организации прерываний. Многоуровневые системы прерываний. Приоритеты

Прерывания – это особые ситуации, которые заставляют процессор приостановить текущую программу и переключиться на выполнение другой, более срочной или важной программы. Причинами прерываний текущей программы могут быть:

- внешний сигнал по входу маскируемых прерываний INTR или немаскируемого прерывания NMI;
- аномальная ситуация, сложившаяся при выполнении конкретной команды и препятствующая нормальному продолжению программы;
- находящаяся в программе команда прерывания INT n.

Первая из указанных выше причин относится к аппаратным прерываниям, а две другие – к программным прерываниям. Отметим, что аппаратные прерывания непредсказуемы и могут возникать в произвольные моменты времени. С помощью аппаратных прерываний осуществляется взаимодействие процессора с периферийными устройствами (клавиатурой, дисками, таймером и т.д.), сообщается о возникновении ошибки на шине или в памяти, об аварийном подключении сети и т.п. Реагируя на внешнее прерывание, процессор должен идентифицировать его источник, сохранить минимальный контекст текущей программы (в частности, адрес возврата) и переключиться на специальную программу – обработчик прерывания (interrupt handler), которым может быть процедура или задача. Действия обработчика прерывания, называемые обслуживанием прерывания, заключается в том, чтобы правильно отреагировать на прерывание от конкретного источника: поместить символ с клавиатуры в буфер, считать сектор с диска, произвести инкремент системных часов и т.п. После обслуживания прерывания процессор возвращается к прерванной программе, и она должна возобновиться так, как будто прерывания не было.

Программные прерывания обычно называются особыми случаями (exceptions), хотя иногда применяется термин «исключение». Особые случаи возникают, например, при нарушении защиты по привилегиям, превышения предела сегмента, выходе за границы массива, деления на нуль и т.д. Как и аппаратные прерывания, предвидеть такие особые случаи практически невозможно. Вместе с тем, особый случай, инициируемый командой INT n, находится под управлением программиста и может использоваться, например, для отладки.

Общая реакция процессора на особый случай аналогична приведенной выше реакции на аппаратное прерывание: необходимо сохранить минимальный контекст программы, идентифицировать источник (причину) особого случая и переключиться на обработчик особого случая (exception handler), которым также может быть процедура или задача. Его действия зависят от условий возникновения особого случая; эти условия приводят к следующей классификации особых случаев.

НАРУШЕНИЕ (fault) – это такой особый случай, который процессор может обнаружить до возникновения фактической ошибки (типичный пример – нарушение правил привилегий или превышения предела сегмента). Очевидно, после обработки нарушения можно продолжить программу, осуществив повторное выполнение (рестарт) виноватой команды.

ЛОВУШКА (trap) – это такой особый случай, который обнаруживается после окончания виноватой команды. После его обработки процессор возобновляет действия с той команды, которая находится после «захваченной» команды. Типичным примером ловушки служат команда прерывания при переполнении и команда INT n. Кроме того, большинство отладочных контрольных точек также интерпретируются как ловушки.

АВАРИЯ (abort) – представляет собой столь серьезную ошибку, что некоторый контекст программы теряется и продолжить ее невозможно. Причину аварии установить нельзя, поэтому осуществить рестарт программы не удастся и ее необходимо прекратить. К авариям относятся аппаратные ошибки, а также несовместимые или недопустимые значения в системных таблицах.

Обработка всех прерываний и особых случаев в процессорах фирмы Intel производится, в общем, одинаково и состоит из двух основных этапов. На первом из них процессор выполняет некоторые «рефлексивные» операции, которые одинаковы для всех прерываний и особых случаев, и которыми программист управлять не может. На втором этапе действует созданный программистом обработчик прерывания или особого случая.

Самое первое действие процессора при распознавании прерывания или особого случая заключается в идентификации его источника. Всем таким источникам назначается идентифицирующий номер от 0 до 255, иногда называемый вектором (vector), который позволяет процессору выбрать нужный обработчик. Некоторые номера назначены аппаратно (например, для всех особых случаев), другие формируют внешние схемы (это все аппаратные прерывания по входу INTR), еще одни зарезервированы фирмой Intel для расширения, а все остальные находятся в распоряжении программиста.

При восприятии аппаратного прерывания процессор выполняет два специальных цикла шины подтверждения прерывания, и прерывающее устройство передает номер прерывания как байт данных. Немаскируемому прерыванию NMI назначен фиксированный номер 2 и не требуются циклы шины подтверждения прерывания.

По номеру процессор выбирает один из 256 обработчиков прерывания, обращаясь к таблице векторов прерываний, находящейся в памяти. Эта состоит из четырехбайтных элементов и начинается с нулевого физического адреса памяти, занимая максимум 1 Кбайт. Каждый элемент таблицы представляет собой полный указатель селектор: смещение точки входа обработчика прерывания. Векторы прерываний должны иметь номера от 32 до 255, так как номера 0-31 зарезервированы фирмой Intel.

6 Средства защиты памяти при организации мультипрограммного режима работы.

При мультипрограммном режиме работы ЭВМ в ее памяти одновременно могут находиться несколько независимых программ. Поэтому необходимы специальные меры по предотвращению или ограничению обращений одной программы к областям памяти, используемым другими программами. Программы могут также содержать ошибки, которые, если этому не воспрепятствовать, приводят к искажению информации, принадлежащей другим программам. Последствия таких ошибок особенно опасны, если разрушению подвергнутся программы операционной системы. Другими словами, надо исключить воздействие программы пользователя на работу программ других пользователей и программ операционной системы. Следует защищать и сами программы от находящихся в них возможных ошибок.

Таким образом, средства защиты памяти должны предотвращать 2

- неразрешенное взаимодействие пользователей друг с другом,
- несанкционированный доступ пользователей к данным,
- повреждение программ и данных из-за ошибок в программах,
- намеренные попытки разрушить целостность системы,
- использование информации в памяти не в соответствии с ее функциональным назначением.

Чтобы воспрепятствовать разрушению одних программ другими, достаточно защитить область памяти данной программы от попыток записи в нее со стороны других программ, а в некоторых случаях и своей программы (защита от записи), при этом допускается обращение других программ к этой области памяти для считывания данных.

В других случаях, например при ограничениях на доступ к информации, хранящейся в системе, необходимо запрещать другим программам любое обращение к некоторой области памяти как на запись, так и на считывание. Такая защита от записи и считывания помогает в отладке программы, при этом осуществляется контроль каждого случая обращения за область памяти своей программы.

Для облегчения отладки программ желательно выявлять и такие характерные ошибки в программах, как попытки использования данных вместо команд или команд вместо данных в собственной программе, хотя эти ошибки могут и не разрушать информацию (несоответствие функционального использования информации).

Если нарушается защита памяти, исполнение программы приостанавливается и вырабатывается запрос прерывания по нарушению защиты памяти.

Защита от вторжения программ в чужие области памяти может быть организована различными методами. Но при любом подходе реализация защиты не должна заметно снижать производительность компьютера и требовать слишком больших аппаратных затрат.

Методы защиты базируются на некоторых классических подходах, которые получили свое развитие в архитектуре современных ЭВМ. К таким методам можно отнести защиту отдельных ячеек, метод граничных регистров, метод ключей защиты.

Защита отдельных ячеек памяти организуется в ЭВМ, предназначенных для работы в системах управления, где необходимо обеспечить возможность отладки новых программ без нарушения функционирования находящихся в памяти рабочих программ, управляющих технологическим процессом. Это может быть достигнуто выделением в каждой ячейке памяти специального "разряда защиты". Установка этого разряда в "1" запрещает производить запись в данную ячейку, что обеспечивает сохранение рабочих программ. Недостаток такого подхода - большая избыточность в кодировании информации из-за излишне мелкого уровня защищаемого объекта (ячейка).

В системах с мультипрограммной обработкой целесообразно организовывать защиту на уровне блоков памяти, выделяемых программам, а не отдельных ячеек.

Метод граничных регистров (рис. 8) заключается во введении двух граничных регистров, указывающих верхнюю и нижнюю границы области памяти, куда программа имеет право доступа.

При каждом обращении к памяти проверяется, находится ли используемый адрес в установленных границах. При выходе за границы обращение к памяти не производится, а формируется запрос прерывания, передающий управление операционной системе. Содержание граничных регистров устанавливается операционной системой при загрузке программы в память.

Модификация этого метода заключается в том, что один регистр используется для указания адреса начала защищаемой области, а другой содержит длину этой области.

Метод граничных регистров, обладая несомненной простотой реализации, имеет и определенные недостатки. Основным из них является то, что этот метод поддерживает работу лишь с непрерывными областями памяти.

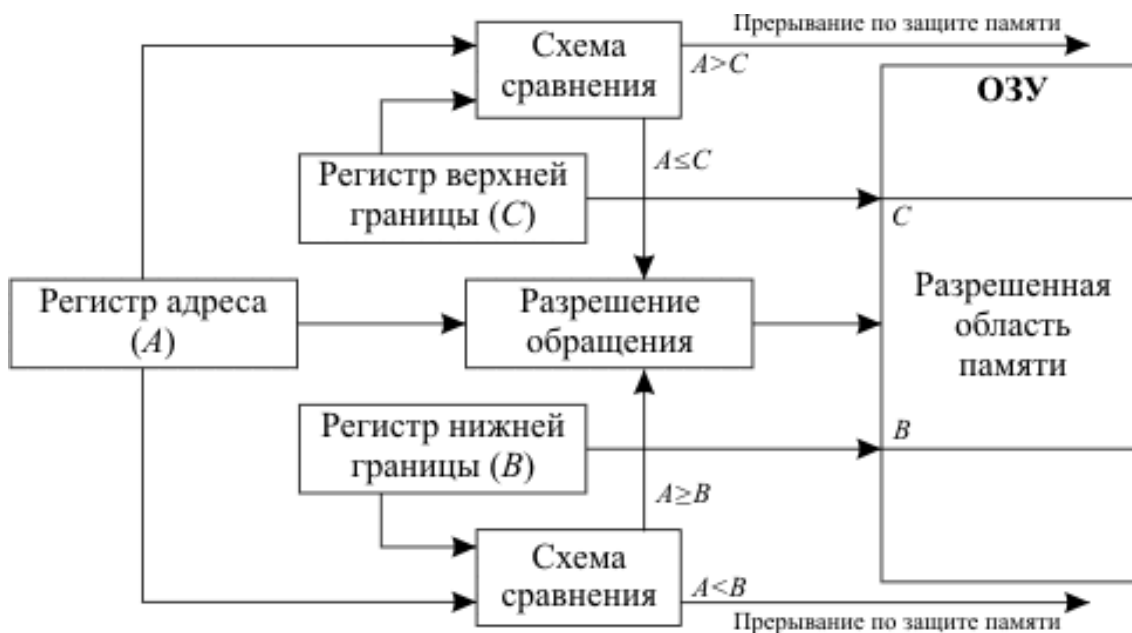


Рисунок 8 - Защита памяти методом граничных регистров

Метод ключей защиты, в отличие от предыдущего, позволяет реализовать доступ программы к областям памяти, организованным в виде отдельных модулей, не представляющих собой единый массив.

Память в логическом отношении делится на одинаковые блоки, например, страницы. Каждому блоку памяти ставится в соответствие код, называемый **ключом защиты памяти**, а каждой программе, принимающей участие в мультипрограммной обработке, присваивается код **ключа программы**. Доступ программы к данному блоку памяти для чтения и записи разрешен, если ключи совпадают (то есть данный блок памяти относится к данной программе) или один из них имеет код 0 (код 0 присваивается программам операционной системы и блокам памяти, к которым имеют доступ все программы: общие данные, совместно используемые подпрограммы и т. п.). Коды ключей защиты блоков памяти и ключей программ устанавливаются операционной системой.

В ключе защиты памяти предусматривается дополнительный разряд режима защиты. Защита действует только при попытке записи в блок, если в этом разряде стоит 0, и при любом обращении к блоку, если стоит 1. Коды ключей защиты памяти хранятся в специальной памяти ключей защиты, более быстродействующей, чем оперативная память.

Функционирование этого механизма защиты памяти поясняется схемой на рисунке 9.



Рисунок 9 - Защита памяти методом ключей защиты

При обращении к памяти группа старших разрядов адреса ОЗУ, соответствующая номеру блока, к которому производится обращение, используется как адрес для выборки из памяти ключей защиты кода ключа защиты, присвоенного операционной системой данному блоку. Схема анализа сравнивает ключ защиты блока памяти и ключ программы, находящийся в регистре слова состояния программы (ССП), и вырабатывает сигнал "Обращение разрешено" или сигнал "Прерывание по защите памяти". При этом учитываются значения режима обращения к ОЗУ (запись или считывание), указываемого триггером режима обращения ТгРО, и режима защиты, установленного в разряде режима обращения (РРО) ключа защиты памяти.

Контрольные вопросы:

1. Какую функцию выполняет устройство управления?
2. Какие блоки включает в себя УУ?
3. Объясните схему выполнения процессором машинных команд.
4. Что такое виртуальная память?
5. В чем разница между физическим и логическим адресом?
6. Что такое интерфейс?
7. Что называют прямым доступом к памяти?
8. Поясните принципы организации управления операционным устройством.
9. Механизм прерываний.
10. Типы прерываний.
11. Назовите способы защиты памяти при организации мультипрограммного режима работы.

Тема 9. Организация ввода - вывода. Вычислительные комплексы

1. Организация ввода - вывода. Интерфейсы, шины расширения ввода – вывода.
2. Способы обмена информацией. Интерфейсы ПК. Параллельные интерфейсы. LPT – порт.
3. Режимы обмена информацией: дуплексный, полудуплексный и симплексный.

1 Организация ввода - вывода. Интерфейсы, шины расширения ввода – вывода

Устройства ввода-вывода (УВВ) входят в состав периферийных устройств ЭВМ. Передача информации с периферийного устройства (ПУ) в ядро ЭВМ называется *операцией ввода*, а передача из ядра ЭВМ в периферийное устройство - *операцией вывода*.

Устройства ввода включают:

1. Клавиатура – устройства ввода электромеханического типа;
2. Сканеры – устройства ввода информации в графическом виде с возможностью последующей программной оцифровки;
3. Графические планшеты – устройства ввода графиков, схем, чертежей с последующей возможностью программной обработки – диджитайзеры;
4. Устройства ввода управляющей информации:
 - мышь;
 - джойстик;
 - световое перо.

Устройства вывода:

1. Экранные пульта:
 - алфавитно-цифровые дисплеи – для вывода алфавитно-цифровой информации;
 - графические дисплеи на ЭЛТ – и текст и графика;
 - векторные дисплеи с режимом регенерации изображения и параметрическим способом получения контура графического образа на экране, предназначены для получения статических и динамических изображений в виде совокупности точек, отрезков векторов и символов;
 - растровые дисплеи – предназначены для получения изображения в виде совокупности точек, отрезков, векторов, символов и закрашенных областей (с помощью матрицы точечных элементов изображения – пикселей);
2. графопостроители – устройства вывода графической информации на бумажные носители;
3. печатающие устройства – принтеры, плоттеры;
4. ВЗУ.

Устройства ввода-вывода информации (УВВ) предназначены для обеспечения связи внешних объектов или человека-оператора с вычислительной системой.

Так как ЭВМ понимает лишь язык физических состояний, кодируемых двоичными числами, то в УВВ обычно производится кодирование (декодирование) информации пересылаемой в ЭВМ (принимаемой от ЭВМ). Данные, закодированные соответствующим образом, передают по линиям связи двумя способами:

- параллельным;
- последовательным.

Когда ЭВМ должна обмениваться данными с удаленными УВВ или другими ЭВМ, то относительно большое расстояние между объектами предопределяет последовательный (побитный) обмен данными. В качестве средства связи чаще всего используют телефонные линии или радиоканал.

При использовании двухпроводной линии связи или радиоканала между УВВ и ЭВМ следует устройства для преобразования данных из параллельного кода в последовательный код и наоборот.

2 Способы обмена информацией. Интерфейсы ПК. Параллельные интерфейсы. LPT – порт

Интерфейс представляет собой совокупность физических средств сопряжения (коннекторов, разъемов), линий и шин, сигналов, электронных схем и алгоритмов, предназначенную для осуществления обмена информацией между устройствами. От характеристик интерфейсов во многом зависят производительность и надежность вычислительной машины.

Время ответа – характеристика системы ввода-вывода – является одним из самых главных показателей производительности ЭВМ. Время ответа определяется временем, прошедшим с момента подачи запроса на ввод-вывод до момента начала передачи данных.

При разработке систем ввода-вывода должны быть решены следующие проблемы:

1. Должна быть обеспечена возможность реализации машин с переменным составом оборудования.
2. Для эффективного использования оборудования ЭВМ должны реализовываться параллельная во времени работа процессора над программой и выполнение периферийными устройствами процедур ввода-вывода.
3. Необходимо стандартизировать программирование операций ввода-вывода для обеспечения их независимости от особенностей периферийного устройства.
4. Необходимо обеспечить автоматическое распознавание и реакцию ядра ЭВМ на многообразие ситуаций, возникающих в ПУ (готовность устройства, различные неисправности и т.п.).

При конструировании ЭВМ широко применяются различные средства унификации.

Средства вычислительной техники проектируются на основе модульного принципа, который заключается в том, что отдельные устройства выполняются

в виде конструктивно законченных модулей, из которых можно собирать ЭВМ в различных конфигурациях.

При обмене между ПУ и ЭВМ используются унифицированные форматы данных. Преобразование унифицированных форматов данных в индивидуальные, приспособленные для отдельных ПУ, производится в самих ПУ. Унификации также подвергают все компоненты интерфейса, а также формат и набор команд процессора для операций ввода-вывода. Унификация распространяется на семейство моделей ЭВМ.

Для обеспечения параллельной работы процессора и периферийных устройств схемы управления вводом-выводом отделяют от процессора. Выполнение общих функций возлагают на общие для групп периферийного оборудования унифицированные устройства - контроллеры прямого доступа к памяти, процессоры ввода-вывода.

BIOS (Basic Input Output System) - базовая система ввода/вывода. Содержит набор основных функций управления стандартными внешними устройствами РС.

Назначение ROM BIOS состоит в выполнении наиболее простых и универсальных услуг, связанных с осуществлением ввода-вывода.

ROM BIOS выполняет три основные функции:

- предоставляет операционной системе аппаратные драйверы и осуществляет сопряжение между материнской платой и остальным РС;
- содержит тест проверки системы, так называемый POST (Power On Self Test), который проверяет при включении РС все важнейшие компоненты;
- содержит программу установки параметров BIOS и аппаратной конфигурации РС – CMOS Setup.

Изменения конфигурации (например, информация о новом винчестере) записываются в специальную область памяти (и оттуда считываются ROM BIOS), называемую CMOS RAM. Эта область памяти (емкостью 100–129 байт) расположена в контроллере периферии 82С206. Для того чтобы записанные значения не были потеряны, контроллер обеспечивается питанием аккумуляторной батареи. Таким образом, информация о конфигурации РС остается в памяти, даже если долго не включать компьютер. Эта аккумуляторная батарея обеспечивает хранение установок CMOS Setup и работу системного таймера, т.е. устройства для отсчета текущего времени.

Интерфейсы внешних устройств. Компьютер, являясь многофункциональным устройством, для расширения возможностей использует различные устройства, подключаемые при помощи интерфейсов. Наиболее часто используемые интерфейсы: USB 2.0, USB 3.0, PS/2, IEEE1394, S/PDIF, Jack 3.5, eSATA, RJ-45, HDMI, D-SUB(VGA), DVI-D, DVI-I, LPT, COM, Thunderbolt. Как правило, внешние интерфейсы располагаются на материнской плате и выведены на задней панели системного блока.



Помимо задней панели материнской платы, внешние интерфейсы могут располагаться на выносных панелях и дополнительных контроллерах.

Таблица 1 - Внешние интерфейсы компьютера.

Порт		Кабель
	USB 2.0 - последовательный интерфейс передачи данных для среднескоростных и низкоскоростных периферийных устройств. До 480 Мбит/с.	
	USB 3.0 - Пропускная способность порта увеличилась до 4,8 Гбит/с. Силой тока увеличилась с 500 мА (USB 2.0) до 900 мА. Кабели обновлённого стандарта физически и функционально совместимы с USB 2.0	
	PS/2 - порт подключения клавиатуры (фиолетовый) и/или мыши (зеленый). В последнее время редко используется, в связи с переходом на USB. Есть переходники для подключения USB клавиатуры или мыши в разъем PS/2.	
	S/PDIF - используется для подключения цифрового оптического аудио кабеля.	
	DVI - предназначен для передачи видеоизображения на цифровые устройства отображения, такие как жидкокристаллические мониторы и проекторы. Виды DVI (см. доп. картинки): DVI-A — только аналоговая передача, DVI-I — аналоговая и цифровая передача, DVI-D — только цифровая передача. Видеокарты с DVI-A не поддерживают стандартные мониторы с DVI-D. Не ошибитесь при выборе монитора!!!	
	VGA (D-SUB) - стандарт аналоговой передачи видеоизображения , постепенно вытесняется DVI.	

	HDMI - интерфейс для мультимедиа высокой чёткости, позволяющий передавать цифровые видеоданные высокого разрешения и многоканальные цифровые аудиосигналы.	
	Jack 3.5 - разъем подключения штекера TRS (<i>Tip, Ring, Sleeve</i> , переводится как <i>Кончик, Кольцо, Гильза</i>). Распространённый разъем для передачи аудиосигнала .	
	IEEE1394 - последовательная высокоскоростная шина, предназначенная для обмена цифровой информацией между компьютером и другими электронными устройствами (видеокамеры, внешние жесткие диски и т.п.).	
	eSATA - последовательный интерфейс обмена данными с накопителями информации . Используется для внешней реализации интерфейса SATA. Может быть использован для горячего подключения жесткого диска (в BIOS необходим режим AHCI). Встречаются также комбинированный разъем eSATA+USB.	
	RJ-45 (LAN) - интерфейс для подключения локальной сети (интернет, интранет).	
	COM - двунаправленный последовательный интерфейс. В настоящее время используется только для подключения специфических устройств (консоль сетевого оборудования, телефонные станции и т.п.) и в последнее время отсутствует на материнских платах.	
	LPT - международный стандарт параллельного интерфейса для подключения периферийных устройств персонального компьютера (принтер, сканер). Практически не используется.	
	Thunderbolt - интерфейс для подключения периферийных устройств. Позволяет достичь скорости до 10 Гб/с на расстоянии до 3 м, при использовании медного кабеля. На текущий момент один из самых быстрых интерфейсов, позиционируемый как замена USB, IEEE1394, DVI, eSATA.	

3 Режимы обмена информацией: дуплексный, полудуплексный и симплексный

Так как ЭВМ понимает лишь язык физических состояний, кодируемых двоичными числами, то в УВВ обычно производится кодирование (декодирование) информации пересылаемой в ЭВМ (принимаемой от ЭВМ). Данные, закодированные соответствующим образом, передают по линиям связи двумя способами:

- параллельным;
- последовательным.

Когда ЭВМ должна обмениваться данными с удаленными УВВ или другими ЭВМ, то относительно большое расстояние между объектами предопределяет последовательный (побитный) обмен данными. В качестве средства связи чаще всего используют телефонные линии или радиоканал.

При использовании двухпроводной линии связи или радиоканала между УВВ и ЭВМ следует устройства для преобразования данных из параллельного кода в последовательный код и наоборот.

Передача данных по линиям связи осуществляется с использованием следующих режимов:

- симплексный;
- полудуплексный;
- дуплексный.

Все периферийные устройства ввода должны коммутироваться с ЭВМ таким образом, чтобы данные, вводимые пользователем, могли не только корректно поступать в компьютер, но и в дальнейшем эффективно обрабатываться. В случае вывода информация, поступающая на монитор, принтер или другие внешние периферийные устройства, должна быть предварительно обработана таким образом, чтобы соответствовать спецификации соответствующих выходных устройств.

В ЭВМ применяются **три режима ввода/вывода**: программно-управляемый ВВ (называемый также программным или нефорсированным ВВ), ВВ по прерываниям (форсированный ВВ) и прямой доступ к памяти. Первый из них характеризуется тем, что инициирование и управление ВВ осуществляется программой, выполняемой процессором, а внешние устройства играют сравнительно пассивную роль и сигнализируют только о своем состоянии, в частности, о готовности к операциям ввода/вывода. Во втором режиме ВВ иницируется не процессором, а внешним устройством, генерирующим специальный сигнал прерывания. Реагируя на этот сигнал готовности устройства к передаче данных, процессор передает управление подпрограмме обслуживания устройства, вызвавшего прерывание. Действия, выполняемые этой подпрограммой, определяются пользователем, а непосредственными операциями ВВ управляет процессор. Наконец, в режиме прямого доступа к памяти, который используется, когда пропускной способности процессора недостаточно, действия процессора приостанавливаются, он отключается от системной шины и не участвует в передачах данных между основной памятью

и быстродействующим ВУ. Заметим, что во всех вышеуказанных случаях основные действия, выполняемые на системной магистрали ЭВМ, подчиняются двум основным принципам.

1. В процессе взаимодействия любых двух устройств ЭВМ одно из них обязательно выполняет активную, управляющую роль и является задатчиком, второе оказывается управляемым, исполнителем. Чаще всего задатчиком является процессор.

2. Другим важным принципом, заложенным в структуру интерфейса, является принцип квитирования (запроса - ответа): каждый управляющий сигнал, посланный задатчиком, подтверждается сигналом исполнителя. При отсутствии ответного сигнала исполнителя в течение заданного интервала времени формируется так называемый тайм-аут, задатчик фиксирует ошибку обмена и прекращает данную операцию.

Связь устройств ЭВМ друг с другом осуществляется с помощью средств сопряжения - интерфейсов.

Контрольные вопросы:

1. Перечислите устройства ввода-вывода информации в ЭВМ.
2. Как происходит ввод и вывод информации?
3. Какие режимы ввода/вывода в ЭВМ вы знаете?
4. Назначение каналов ввода-вывода, их характеристики.
5. Укажите современные интерфейсы внешних устройств.
6. Охарактеризуйте дуплексный, полудуплексный и симплексный режимы обмена информацией.

Тема 10. Интерфейсы передачи данных

1. Последовательные интерфейсы. СОМ – порт.
2. Способы последовательной передачи. Асинхронный и синхронный обмен информацией.
3. Режимы передачи данных. Управление потоком передачи.
4. Контроллеры внешних устройств.
5. Порты ввода – вывода. Обмен информацией при прерываниях.

1 Последовательные интерфейсы. СОМ – порт

Универсальный внешний последовательный интерфейс — СОМ-порт (Communications Port — коммуникационный порт) присутствует в РС начиная с первых моделей. Этот порт обеспечивает асинхронный обмен по стандарту RS-232С. СОМ-порты реализуются на микросхемах универсальных асинхронных приемопередатчиков - (UART), совместимых с семейством i8250/16450/16550. Они занимают в пространстве ввода-вывода по 8 смежных 8-битных регистров и могут располагаться по стандартным базовым адресам 3F8h (СОМ1), 2F8h (СОМ2), 3E8h (СОМ3), 2E8h (СОМ4). Порты могут вырабатывать аппаратные прерывания IRQ4 (обычно используются для СОМ1 и СОМ3) и IRQ3 (для СОМ2 и СОМ4). С внешней стороны порты имеют линии последовательных данных передачи и приема, а также набор сигналов управления и состояния, соответствующий стандарту RS-232С. СОМ-порты имеют внешние разъемы (male — «папа») DB25P или DB9P, выведенные на заднюю панель компьютера. Характерной особенностью интерфейса является применение «не ТТЛ» сигналов — все внешние сигналы порта двуполярные. Гальваническая развязка отсутствует — схемная земля подключаемого устройства соединяется со схемной землей компьютера. Скорость передачи данных может достигать 115 200 бит/с.

Компьютер может иметь до четырех последовательных портов СОМ 1-СОМ4 (для машин класса АТ типично наличие двух портов) с поддержкой на уровне BIOS. Сервис BIOS Int 14h обеспечивает инициализацию порта, ввод и вывод символа (не используя прерываний) и опрос состояния. Через Int 14h скорость передачи программируется в диапазоне 110-9600 бит/с (меньше, чем реальные возможности порта). Для повышения производительности широко используется взаимодействие программ с портом на уровне регистров, для чего требуется совместимость аппаратных средств СОМ-порта с программной моделью 18250/16450/16550.

Название порта указывает на его основное назначение — подключение коммуникационного оборудования (например, модема) для связи с другими компьютерами, сетями и периферийными устройствами. К порту могут непосредственно подключаться и периферийные устройств с последовательным интерфейсом: принтеры, плоттеры, терминалы и другие. СОМ-порт широко

используется для подключения мыши, а также организации непосредственной связи двух компьютеров. К СОМ-порту подключают и электронные ключи. Синхронный обмен в РС поддерживают лишь специальные адаптеры, например SDLC или V.35.

Практически все современные системные платы (еще начиная с РС1-плат для процессоров 486) имеют встроенные адаптеры двух СОМ-портов.

2 Способы последовательной передачи. Асинхронный и синхронный обмен информацией

Последовательная передача данных может осуществляться в асинхронном или синхронном режимах. При *асинхронной* передаче каждому байту предшествует *старт-бит*, сигнализирующий приемнику о начале посылки, за которым следуют *биты данных* и, возможно, *бит паритета* (четности). Завершает посылку *стоп-бит*, гарантирующий паузу между посылками (рис. 1). Старт-бит следующего байта посылается в любой момент после стоп-бита, то есть между передачами возможны паузы произвольной длительности. Старт-бит, имеющий всегда строго определенное значение (логический 0), обеспечивает простой механизм синхронизации приемника по сигналу от передатчика. Подразумевается, что приемник и передатчик работают на одной скорости обмена. Внутренний генератор синхронизации приемника использует счетчик-делитель опорной частоты, обнуляемый в момент приема начала старт-бита. Этот счетчик генерирует внутренние стробы, по которым приемник фиксирует последующие принимаемые биты. В идеале стробы располагаются в середине битовых интервалов, что позволяет принимать данные и при незначительном рассогласовании скоростей приемника и передатчика. Очевидно, что при передаче 8 бит данных, одного контрольного и одного стоп-бита предельно допустимое рассогласование скоростей, при котором данные будут распознаны верно, не может превышать 5%. С учетом фазовых искажений и дискретности работы внутреннего счетчика синхронизации реально допустимо меньшее отклонение частот. Чем меньше коэффициент деления опорной частоты внутреннего генератора (чем выше частота передачи), тем больше погрешность привязки стробов к середине битового интервала, и требования к согласованности частот становятся более строгими. Чем выше частота передачи, тем больше влияние искажений фронтов на фазу принимаемого сигнала. Взаимодействие этих факторов приводит к повышению требований к согласованности частот приемника и передатчика с ростом частоты обмена.



Рисунок 1 - Формат асинхронной передачи

Формат асинхронной посылки позволяет выявлять возможные *ошибки передачи*:

- Если принят перепад, сигнализирующий о начале посылки, а по стробу старт-бита зафиксирован уровень логической единицы, старт-бит считается ложным и приемник снова переходит в состояние ожидания. Об этой ошибке приемник может и не сообщать.

- Если во время, отведенное под стоп-бит, обнаружен уровень логического нуля, фиксируется ошибка стоп-бита.

- Если применяется контроль четности, то после посылки бит данных передается *контрольный бит*. Этот бит дополняет количество единичных бит данных до четного или нечетного в зависимости от принятого соглашения. Прием байта с неверным значением контрольного бита приводит к фиксации ошибки.

Контроль формата позволяет обнаруживать обрыв линии: при этом принимаются логический нуль, который сначала трактуется как старт-бит, и нулевые биты данных, потом срабатывает контроль стоп-бита.

Для асинхронного режима принят ряд *стандартных скоростей обмена*: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 и 115200 бит/с. Иногда вместо единицы измерения "бит/с" используют "бод" (baud), но при рассмотрении двоичных передаваемых сигналов это некорректно. В бодах принято измерять частоту изменения состояния линии, а при недвоичном способе кодирования (широко применяемом в современных модемах) в канале связи скорости передачи бит (бит/с) и изменения сигнала (бод) могут отличаться в несколько раз (подробнее см. в приложении А).

Количество *бит данных* может составлять 5, 6, 7 или 8 (5- и 6-битные форматы распространены незначительно). Количество *стоп-бит* может быть 1, 1,5 или 2 ("полтора бита" означает только длительность стопового интервала).

Асинхронный обмен в РС реализуется с помощью *COM-порта* с использованием протокола *RS-232C*.

Синхронный режим передачи предполагает постоянную активность канала связи. Посылка начинается с синхробайта, за которым сразу же следует поток информационных бит. Если у передатчика нет данных для передачи, он заполняет паузу непрерывной посылкой байтов синхронизации. Очевидно, что при передаче больших массивов данных накладные расходы на синхронизацию в данном режиме будут ниже, чем в асинхронном. Однако в синхронном

режиме необходима внешняя синхронизация приемника с передатчиком, поскольку даже малое отклонение частот приведет к искажению принимаемых данных. Внешняя синхронизация возможна либо с помощью отдельной линии для передачи сигнала синхронизации, либо с использованием самосинхронизирующего кодирования данных, при котором на стороне приемника из принятого сигнала могут быть выделены импульсы синхронизации. В любом случае синхронный режим требует дорогих линий связи или оконечного оборудования. Для РС существуют специальные платы - адаптеры SDLC, поддерживающие синхронный режим обмена. Они используются в основном для связи с большими машинами (mainframes) IBM и мало распространены. Из синхронных адаптеров в настоящее время применяются адаптеры интерфейса V.35.

К примеру, что значит RS в сокращениях типа RS-232, RS-485, RS-422.. RS - это всего навсего Recommended Standard (рекомендованный стандарт). Ключевое слово тут - "рекомендованный", означающее, что эти стандарты никогда никем не были приняты (в противоположность таким стандартам, как IEEE-1284 или IEEE-1394), они были просто "рекомендованны". Естественно, это позволяет производителям гордиться тем, что горазд (например, питание по 9-му пину в RS-232 вовсе не оговорено стандартом, однако широко используется) и называется это стандартом. Далее, все RS-протоколы можно приблизительно разделить на полудуплексные (half-duplex) и дуплексные (full-duplex). Правда, деление такое не совсем точно, т.к. тот же RS-485 может быть и полудуплексным (два провода) и дуплексным (четыре провода), они так и называются - 2-wire (2-проводный) RS-485 и 4-wire (4-проводный) RS-485. RS-485 именно полудуплексный, RS-232 - дуплексный протокол.

3 Режимы передачи данных. Управление потоком передачи

Передача данных (обмен данными, цифровая передача, цифровая связь) — физический перенос данных (цифрового битового потока) в виде сигналов от точки к точке или от точки к нескольким точкам средствами электросвязи по каналу передачи данных, как правило, для последующей обработки средствами вычислительной техники. Примерами подобных каналов могут служить медные провода, ВОЛС, беспроводные каналы передачи данных или запоминающее устройство.

Передача данных может быть аналоговой или цифровой (то есть поток двоичных сигналов), а также модулирован посредством аналоговой модуляции, либо посредством цифрового кодирования.

Хотя аналоговая связь является передачей постоянно меняющегося цифрового сигнала, цифровая связь является непрерывной передачей сообщений. Сообщения представляют собой либо последовательность импульсов, означающую линейный код (в полосе пропускания), либо ограничивается набором непрерывно меняющейся формы волны, используя метод цифровой модуляции. Такой способ модуляции и соответствующая ему демодуляция осуществляются модемным оборудованием.

Передаваемые данные могут быть цифровыми сообщениями, идущими из источника данных, например, из компьютера или от клавиатуры. Это может быть и аналоговый сигнал — телефонный звонок или видеосигнал, оцифрованный в битовый поток, используя импульсно-кодирующую модуляцию (PCM) или более расширенные схемы кодирования источника (аналого-цифровое преобразование и сжатие данных). Кодирование источника и декодирование осуществляется кодеком или кодирующим оборудованием.

Управление потоком передачи данных (англ. *Flow Control*) — в компьютерных сетях, механизм, который притормаживает передатчик данных при неготовности приёмника.

Различают три основных способа:

- аппаратный, при котором сигналы «готов/занят» передаются по отдельным физическим линиям связи. Наиболее известна такая реализация в интерфейсе RS-232.

- программный, при котором программный флажок «готов/занят» взводится и сбрасывается вставкой в поток данных специальной уникальной последовательности (XOn/XOff). Применяется в программных драйверах интерфейса RS-232 как альтернатива аппаратному контролю потока в случаях неполного соединительного кабеля.

- протокольный, при котором программный флажок «готов/занят» взводится и сбрасывается специальными соглашениями в рамках протокола обмена данными. На сегодня является практически единственным применяемым способом контроля потока. Наиболее известный пример — реализация контроля потока в протоколе TCP методом скользящего окна.

Совокупность средств, необходимых для эффективной и экономичной передачи блоков данных в сети. Целью управления трафиком является равномерное распределение нагрузки по всем сегментам сети. Средства управления трафиком осуществляют:

- бесперебойную доставку данных адресатам;
- проверку фактической загрузки каналов и производительности сети;
- управление очередями.

Стандарт 802.3x поддержка полнодуплексной связи; совместимость с DIX

4 Контроллеры внешних устройств

Очевидно, все необходимые действия должна уметь выполнять в определенной последовательности (по определенной логике) некоторая схема управления — своя для каждого устройства ввода или вывода. Действительно, такие схемы есть и называются они контроллерами внешних устройств. Именно они опознают в процессе обмена адреса регистра данных и регистра состояния конкретного устройства, они принимают (или передают) данные на свой внутренний регистр (или из него), взаимодействуют с исполнительной схемой внешних устройств и формируют необходимые сигналы готовности, ошибок и т.п. в соответствующих разрядах регистра состояния, доступного для процессора через магистраль. Такие контроллеры вызывают при необходимос-

ти прерывание работы процессора в соответствии с правилами, установленными на магистрали.

Однако логика обмена микропроцессора с контроллерами внешних устройств оказывается весьма разветвленной, более того, не всякий обмен заканчивается за одно обращение к контроллеру, и поэтому кроме аппаратной части в обмене, как правило, участвует и программа. Иногда необходимая программная часть реализуется простыми действиями всего в несколько команд, но чаще в ней предусматриваются весьма разнообразные действия в зависимости от возможных ситуаций. Такие вспомогательные программы получили специальное название - драйверы внешних устройств, а поскольку с внешними устройствами работают практически все прикладные программы, драйверы, как правило, включают в состав операционных систем. В этом случае любая прикладная программа, обратившись к драйверу, передает ему всю работу, обеспечивающую правильное взаимодействие с внешними устройствами в любой ситуации. Как результат, драйвер либо производит необходимый обмен, либо извещает программу о неудачном исходе.

Контроллеры внешних устройств это электронные блоки, выполненные в виде печатных плат, вставляемые в предназначенные для них разъёмы на материнской плате. Они выполняют функции по передаче управляющих и информационных сигналов с материнской платы на внешнее устройство и обратно. Контроллеры с одинаковыми функциями могут иметь разный тип разъёма (ISA, EISA, PCI, PCMCIA), что следует учитывать при приобретении внешних устройств и контроллеров к ним. Количество разъёмов для контроллеров каждого типа на материнской плате ограничено.

Стандартными являются контроллеры последовательного (COM) и параллельного (LP) портов, мощного последовательного порта (USB), а также, в некоторых компьютерах, мощного параллельного порта (SCSI). На большинстве материнских плат смонтированы контроллеры жёстких дисков (IDE, DMA, SUPER DMA, SCSI) и флоппи дисков.

Параллельные порты применяются для подключения к компьютеру внешних устройств с высоким быстродействием, а также для установления прямого кабельного соединения между компьютерами на расстояние до 2.5 м. Через параллельный порт к компьютеру подключаются такие устройства, как принтеры, графопостроители, некоторые сканеры, Zip диски.

Через COM порт USB порт к компьютеру подключаются такие внешние устройства, как мышь, модем, некоторые принтеры, а также устанавливается прямое кабельное соединение между компьютерами на расстояние до 10 м.

SCSI порт это быстродействующий параллельный порт. Через SCSI порт к компьютеру подключаются очень быстродействующие внешние устройства, а также устанавливается прямое кабельное соединение между компьютерами на расстояние до 1.5 м. Через SCSI порт к компьютеру подключаются такие устройства, как высокопроизводительные лазерные принтеры, высокоточные сканеры.

5 Порты ввода – вывода. Обмен информацией при прерываниях

Ввод-вывод через порты (англ. I/O ports) — схемотехническое решение, организующее взаимодействие процессора и устройств ввода-вывода. Противоположность **вводу-выводу через память**.

Во многих моделях процессоров ввод-вывод организуется теми же функциями, что и чтение-запись в память — так называемый «ввод-вывод через память». Соответственно, схемотехнически устройства ввода-вывода располагаются на шине памяти, и часть адресов памяти направляются на ввод-вывод. В процессорах Intel, микроконтроллерах AVR и некоторых других существуют отдельные команды для ввода-вывода — IN и OUT — и, соответственно, отдельное адресное пространство: в процессорах Intel — от 0000_{16} до $FFFF_{16}$.

Порты ввода-вывода создаются в системном оборудовании, которое циклически декодирует управляющие, адресные и контакты данных процессора. Затем порты настраиваются для обеспечения связи с периферийными устройствами ввода-вывода.

Одни порты используются для передачи данных (например, приём данных от клавиатуры или чтение времени системных часов), другие — для управления периферийными устройствами (команда чтения данных с диска). Исходя из этого порт ввода-вывода может быть портом только для ввода, только вывода, а также двунаправленным портом.

Ввод-вывод через память никак не связан с прямым доступом к памяти; ПДП — отдельное схемное решение, связывающее шину ввода-вывода с контроллером памяти и разгружающее процессор на крупных операциях ввода-вывода. В машине с ПДП, чтобы записать блок памяти, например, на диск, надо сформировать этот блок в памяти, а затем операциями ввода-вывода (либо через порт, либо через специальный адрес памяти, в зависимости от архитектуры) отправить команду «Начать запись». Когда запись будет закончена, устройство каким-то образом предупредит процессор об этом (например, прерыванием). В частности, известные любому знакомому с DOS три параметра SoundBlaster — порт, DMA и IRQ — указывают, как передавать звуковой плате команды, как она будет брать звуковую волну из памяти и как плата сообщит процессору, что отрезок волны проигран.

Порты или память?

Преимущества портов:

- Возможна совсем другая схемотехническая организация ввода-вывода.
- Человек, читающий ассемблерный листинг, сразу же видит, что это работа не с памятью, а с внешним устройством.
- Всё адресное пространство машины можно пустить на ОЗУ, без всяких «дыр» наподобие UMB.
- Для машин низкой разрядности, у которых адрес в памяти задаётся регистровой парой, а на порт хватает и одного регистра, ускоряется работа с внешними устройствами.

Преимущества памяти:

- Упрощение конструкции процессора.
- Более широкий набор возможных инструкций: все инструкции, способные записать данные в память, в том числе автоинкрементные и SIMD, могут заниматься вводом-выводом.
- Удобнее работа с функциями, оперирующими большими объёмами данных (например, чтение-запись на диск, в видеоадаптер).

Порты в архитектуре IA-32

Процессор позволяет осуществлять ввод-вывод как через память, так и через порты. Доступ к портам осуществляется особыми командами IN, INS, INSB, INSW, INSD, OUT, OUTS, OUTSB, OUTSW и OUTSD; доступ к памяти — обычными функциями работы с памятью (MOV, ADD, LODSB и прочими). Два последовательных порта могут объединяться в один двухбайтовый порт, четыре — в один четырёхбайтовый. Как и с памятью, для максимальной производительности номера портов должны быть выровнены соответственно по 2- и 4-байтовой границе. Если требуется записывать данные в два соседних порта в определённом порядке, этого нельзя делать многобайтовыми командами (на выровненных портах ввод-вывод происходит параллельно, на невыровненных — порядок вызова не определён).

Процессор умеет проецировать порты в память; при этом процессор гарантирует, что операция ввода-вывода через порт завершится до того, как начнёт выполняться следующая команда. С вводом-выводом через память такой гарантии нет. Впрочем, процессор не проверяет ошибок чётности на портах, поэтому в системах высокой надёжности программист должен быть готов к ошибкам чётности.

Физически адрес порта подаётся через ту же адресную шину, что и адрес памяти. В процессорах Intel архитектуры x86 (до Pentium) была линия MI/O# (1 — память, 0 — порт); в более поздних эту функцию выполняют линии запроса команды.

Обмен по прерываниям используется в том случае, когда необходима реакция микропроцессорной системы на какое-то внешнее событие. Внешним событием может быть:

- нажатие на клавишу клавиатуры
- приход по локальной сети пакета данных.

Компьютер должен отреагировать на эти события. Например

- выводом символа на экран;
- чтением и обработкой принятого по сети пакета.

В общем случае организовать реакцию на внешнее событие можно тремя различными путями:

- с помощью постоянного программного контроля факта наступления события (так называемый метод опроса флага или polling);
- с помощью прерывания, то есть насильственного перевода процессора с выполнения текущей программы на выполнение экстренно необходимой программы;

- с помощью прямого доступа к памяти, то есть без участия процессора при его отключении от системной магистрали.

Контрольные вопросы:

1. Где в ЭВМ применяется универсальный внешний последовательный интерфейс — СОМ-порт?
2. В каких режимах может осуществляться последовательная передача данных?
3. Охарактеризуйте особенности асинхронной и синхронной передачи данных.
4. Как осуществляется управление потоком передачи данных?
5. Назначение контроллеров внешних устройств.
6. Чем отличается ввод-вывод через порты от ввода-вывода через память?
7. Что означает термин «прямой доступ к памяти»?

Тема 11. Распределенные системы обработки данных

1. Распределенные системы обработки данных.
2. Вычислительные комплексы. Классификация.
3. Многомашинные вычислительные комплексы (системы). Организация функционирования многомашинной вычислительной системы (ММВС). Виды связей в ММВС.
4. Мультипроцессорные компьютерные системы (МКС). Типы структурной организации и основы проектирования МКС.

1 Распределенные системы обработки данных

Классифицируются СОД в зависимости от способа построения. СОД, построенные на базе вычислительных комплексов и систем, образуют класс *сосредоточенных (централизованных) систем*, в которых вся обработка реализуется ЭВМ, вычислительным комплексом или специализированной системой. В настоящее время все большее распространение получают *распределенные системы*, к которым относятся системы телеобработки и вычислительные сети. В них процессы обработки данных рассредоточены по многим компонентам. При этом системы телеобработки считаются распределенными в некоторой степени условно, поскольку основные функции обработки данных здесь реализуются централизованно – в одной ЭВМ или вычислительном комплексе.

В распределенных системах из-за значительности расстояний между компонентами применяются последовательные и связные интерфейсы, которые исключают возможность передачи сигналов прерывания между сопрягаемыми устройствами и требуют представления данных в виде сообщений, передаваемых с помощью операций ввода-вывода.

2 Вычислительные комплексы. Классификация

Вычислительные комплексы. Начиная с 60-х годов для повышения надежности и производительности СОД, несколько ЭВМ связывались между собой, образуя *многомашинный вычислительный комплекс*.

В ранних многомашинных комплексах связь между ЭВМ обеспечивалась через общие внешние запоминающие устройства – накопители на магнитных дисках (НМД) или магнитных лентах (НМЛ) (рис 1,1, *а*), т.е. за счет доступа к общим наборам данных. Такая связь называется *косвенной* и оказывается эффективной только в том случае, когда ЭВМ взаимодействуют достаточно редко, например, при отказе одной из ЭВМ или в моменты начала и окончания обработки данных. Более оперативное взаимодействие ЭВМ достигается за счет *прямой связи* через адаптер, обеспечивающий обмен данными между каналами ввода – вывода ЧКВВ) двух ЭВМ (рис. 1.1, *б*) и передачу сигналов прерывания. За счет этого создаются хорошие условия для координации процессов обработки данных и повышается оперативность обмена данными, что

позволяет вести параллельно процессы обработки и существенно увеличивать производительность СОД. В настоящее время многомашинные вычислительные комплексы широко используются для повышения надежности и производительности СОД.

В многомашинных вычислительных комплексах взаимодействие процессов обработки данных обеспечивается только за счет обмена сигналами прерывания и передачи данных через адаптеры канал – канал или общие внешние запоминающие устройства. Лучшие условия для взаимодействия процессов – когда все процессоры имеют доступ ко всему объему данных, хранимых в оперативных запоминающих устройствах (ОЗУ), и могут взаимодействовать со всеми периферийными устройствами комплекса. Вычислительный комплекс, содержащий несколько процессоров с общей оперативной памятью и периферийными устройствами, называется *многопроцессорным*. Принцип построения таких комплексов иллюстрируется рис. 1. Процессоры, модули оперативной памяти (МП) и каналы ввода–вывода, к которым подключены периферийные устройства (ПУ), объединяются в единый комплекс с помощью средств коммутации, обеспечивающих доступ каждого процессора к любому модулю оперативной памяти и каналу ввода–вывода, а также возможность передачи данных между последними. В многопроцессорном комплексе отказы отдельных устройств влияют на работоспособность СОД в меньшей степени, чем в многомашинном, т.е. многопроцессорные комплексы обладают большей устойчивостью к отказам. Каждый процессор имеет непосредственный доступ ко всем данным, хранимым в общей оперативной памяти, и к периферийным устройствам, что позволяет параллельно обрабатывать не только независимые задачи, но и блоки одной задачи.

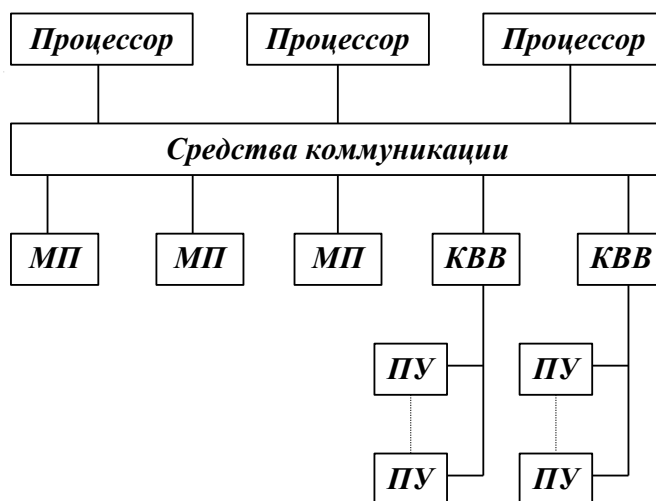


Рисунок 1 - Многопроцессорный вычислительный комплекс

Многомашинные и многопроцессорные вычислительные комплексы рассматриваются как базовые средства для создания СОД различного назначения. Поэтому в состав вычислительного комплекса принято включать только технические средства и общесистемное (базовое), но не прикладное

программное обеспечение, связанное с конкретной областью применения комплекса. Таким образом, *вычислительный комплекс* – совокупность технических средств, включающих в себя несколько ЭВМ или процессоров, и общесистемного (базового) программного обеспечения.

Вычислительные системы. СОД, настроенная на решение задач, конкретной области применения, называется *вычислительной системой*. Вычислительная система включает в себя технические средства и программное обеспечение, ориентированные на решение определенной совокупности задач. Существует два способа ориентации. Во-первых, вычислительная система может строиться на основе ЭВМ или вычислительного комплекса общего применения и ориентация системы обеспечивается за счет программных средств – прикладных программ и, возможно, операционной системы. Во-вторых, ориентация на заданный класс задач может достигаться за счет использования специализированных ЭВМ и вычислительных комплексов. В этом случае удастся при умеренных затратах оборудования добиться высокой производительности. Специализированные вычислительные системы наиболее широко используются при решении задач векторной и матричной, алгебры, а также связанных с интегрированием дифференциальных уравнений, обработкой изображений, распознаванием образов и т. д.

Вычислительные системы, построенные на основе специализированных комплексов, начали интенсивно разрабатываться с конца 60-х годов. В таких системах использовались процессоры со специализированными системами команд, конфигурация комплексов жестко ориентировалась на конкретный класс задач. В последнее десятилетие начались исследования и разработки адаптивных вычислительных систем, гибко приспособляющихся к решаемым задачам. Адаптация вычислительной системы с целью приспособления ее к структуре реализуемого алгоритма достигается за счет изменения конфигурации системы. При этом соединения между процессорами, а также модулями памяти и периферийными устройствами устанавливаются динамически в соответствии с потребностями задач, обрабатываемых системой в текущий момент времени. В связи с этим адаптивные вычислительные системы иначе называются системами с динамической структурой. За счет адаптации достигается высокая производительность в широком классе задач и обеспечивается устойчивость системы к отказам. Поэтому адаптивные системы рассматриваются как одно из перспективных направлений развития систем обработки данных.

Системы телеобработки. Уже первоначальное применение СОД для управления производством, транспортом и материально-техническим снабжением показало, что эффективность систем можно значительно повысить, если обеспечить ввод данных в систему непосредственно с мест их появления и выдачу результатов обработки к местам их использования. Для этого необходимо связать СОД и рабочие места пользователей с помощью каналов связи. Системы, предназначенные для обработки данных, передаваемых по каналам связи, называются *системами телеобработки данных*.

Пользователи (абоненты) взаимодействуют с системой посредством терминалов (абонентских пунктов), подключаемых через каналы связи к средствам обработки данных – ЭВМ или вычислительному комплексу. Данные передаются по каналам связи в форме сообщений – блоков данных, несущих в себе кроме собственно данных служебную информацию, необходимую для управления процессами передачи и защиты данных от искажений. Программное обеспечение систем телеобработки содержит специальные средства, необходимые для управления техническими средствами, установления связи между ЭВМ и абонентами, передачи данных между ними и организации взаимодействия пользователей с программами обработки данных.

Телеобработка данных значительно повышает оперативность информационного обслуживания пользователей и наряду с этим позволяет создавать крупномасштабные системы, обеспечивающие доступ широкого круга пользователей к данным и процедурам их обработки.

3 Многомашинные вычислительные комплексы (системы). Организация функционирования многомашинной вычислительной системы (ММВС). Виды связей в ММВС

Многомашинный вычислительный комплекс (ММВК) — комплекс, включающий в себя две или более ЭВМ (каждая из которых имеет процессор, ОЗУ, набор периферийных устройств и работает под управлением собственной операционной системы), связи между которыми обеспечивают выполнение функций, возложенных на комплекс.

Цели, которые ставятся при объединении ЭВМ в комплекс, могут быть различными, и они определяют характер связей между ЭВМ. Чаще всего основной целью создания ММВК является или увеличение производительности, или повышение надежности, или одновременно и то и другое. Однако при достижении одних и тех же целей связи между ЭВМ могут существенно различаться.

По характеру связей между ЭВМ комплексы можно разделить на три типа: косвенно-, или слабосвязанные; прямосвязанные; спутниковые.

В *косвенно-, или слабосвязанных, комплексах* ЭВМ связаны друг с другом только через внешние запоминающие устройства (ВЗУ). Для обеспечения таких связей используются устройства управления ВЗУ с двумя и более входами. Структурная схема такого ММВК приведена на рис. 2. Заметим, что здесь и далее для простоты приводятся схемы для двухмашинных комплексов. При трех и более ЭВМ комплексы строятся аналогичным образом. В косвенносвязанных комплексах связь между ЭВМ осуществляется только на информационном уровне. Обмен информацией осуществляется в основном по принципу «почтового ящика», т. е. каждая из ЭВМ помещает в общую внешнюю память информацию, руководствуясь собственной программой, и соответственно другая ЭВМ принимает эту информацию, исходя из своих потребностей. Такая организация связей обычно используется в тех случаях, когда ставится задача повысить надежность комплекса путем

резервирования ЭВМ. В этом случае ЭВМ, являющаяся основной, решает заданные задачи, выдает результаты и постоянно оставляет в общем ВЗУ всю информацию, необходимую для продолжения решения с любого момента времени. Вторая ЭВМ, являющаяся резервной, может находиться в состоянии ожидания, с тем, чтобы в случае выхода из строя основной ЭВМ по сигналу оператора начать выполнение функций, используя информацию, хранимую в общем ВЗУ основной ЭВМ.

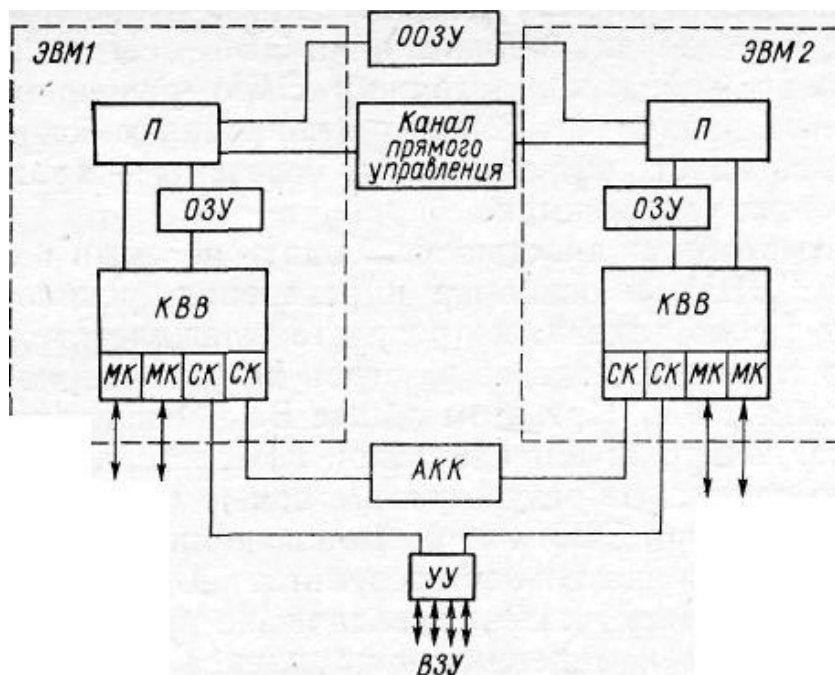


Рисунок 2 - Связи между ЭВМ и ММВК

При такой связи может быть несколько способов организации работы комплекса.

1. Резервная ЭВМ находится в выключенном состоянии (не нагруженный резерв) и включается только при отказе основной ЭВМ. Естественно, для того чтобы резервная ЭВМ начала выдавать результаты вместо основной, потребуется определенное время, которое определяется временем, необходимым для включения ЭВМ, вхождением ее в режим, а также временем, отводимым для проверки ее исправности. Это время может быть достаточно большим. Такая организация возможна, когда система, в которой работает ЭВМ, не критична по отношению к некоторым перерывам или остановкам в процессе решения задач. Это обычно имеет место в случаях, когда ЭВМ не выдает управляющую информацию.

2. Резервная ЭВМ находится в состоянии полной готовности и в любой момент может заменить основную ЭВМ (нагруженный резерв), причем либо не решает никаких задач, либо работает в режиме самоконтроля, решая контрольные задачи. В этом случае переход в работе от основной к резервной ЭВМ может осуществляться достаточно быстро, практически без перерыва в выдаче результатов. Однако, следует заметить, что основная ЭВМ обновляет в общем ВЗУ информацию, необходимую для продолжения

решения, не непрерывно, а с определенной дискретностью, поэтому резервная ЭВМ начинает решать задачи, возвращаясь на некоторое время назад. Такая организация допустима и в тех случаях, когда ЭВМ работает непосредственно в контуре управления, а управляемый процесс достаточно медленный и возврат во времени не оказывает заметного влияния.

При организации работы по первому и второму вариантам ЭВМ используются нерационально: одна ЭВМ всегда простаивает. Простоев можно избежать, загружая ЭВМ решением каких-то вспомогательных задач, не имеющих отношения к основному процессу. Это повышает эффективность системы — производительность практически удваивается.

3. Для того чтобы полностью исключить перерыв в выдаче результатов, обе ЭВМ, и основная и резервная, решают одновременно одни и те же задачи, но результаты выдает только основная ЭВМ, а в случае выхода ее из строя результаты начинает выдавать резервная ЭВМ. При этом общее ВЗУ используется только для взаимного контроля. Иногда такой комплекс дополняется устройством для сравнения результатов с целью контроля. Если при этом используются три ЭВМ, то возможно применение метода голосования, когда окончательный результат выдается только при совпадении результатов решения задачи не менее чем от двух ЭВМ. Это повышает и надежность комплекса в целом, и достоверность выдаваемых результатов. Разумеется, в этом варианте высокая надежность и оперативность достигается весьма высокой ценой — увеличением стоимости системы.

Следует обратить внимание, что при любой организации работы в слабосвязанном ММВК переключение ЭВМ осуществляется либо по командам оператора, либо с помощью дополнительных средств, осуществляющих контроль исправности ЭВМ и вырабатывающих необходимые сигналы. Кроме того, быстрый переход в работе с основной на резервную ЭВМ возможен лишь при низкой эффективности использования оборудования.

Существенно большей гибкостью обладают прямосвязанные ММВК. В прямосвязанных комплексах существуют три вида связей: общее ОЗУ (ООЗУ); прямое управление, иначе связь процессор (П)—процессор; адаптер канал—канал (АКК).

Связь через общее ОЗУ гораздо сильнее связи через ВЗУ. Хотя первая связь также носит характер информационной связи и обмен информацией осуществляется по принципу «почтового ящика», однако, вследствие того что процессоры имеют прямой доступ к ОЗУ, все процессы в системе могут протекать с существенно большей скоростью, а разрывы в выдаче результатов при переходах с основной ЭВМ на резервную сокращаются до минимума. Недостаток связи через общее ОЗУ заключается в том, что при выходе из строя ОЗУ, которое является сложным электронным устройством, нарушается работа всей системы. Чтобы этого избежать, приходится строить общее ОЗУ из нескольких модулей и резервировать информацию. Это, в свою очередь, приводит к усложнению организации вычислительного процесса в комплексе и, в конечном счете, к усложнению операционных систем. Следует

отметить также и то, что связи через общее ОЗУ существенно дороже, чем через ВЗУ.

Непосредственная связь между процессорами — канал прямого управления — может быть не только информационной, но и командной, т. е. по каналу прямого управления один процессор может непосредственно управлять действиями другого процессора. Это, естественно, улучшает динамику перехода от основной ЭВМ к резервной, позволяет осуществлять более полный взаимный контроль ЭВМ. Вместе с тем передача сколько-нибудь значительных объемов информации по каналу прямого управления нецелесообразна, так как в этом случае решение задач прекращается: процессоры ведут обмен информацией.

Связь через адаптер канал - канал в значительной степени устраняет недостатки связи через общее ОЗУ и вместе с тем почти не уменьшает возможностей по обмену информацией между ЭВМ по сравнению с общим ОЗУ. Сущность этого способа связи заключается в том, что связываются между собой каналы двух ЭВМ с помощью специального устройства — адаптера. Обычно это устройство подключается к селекторным каналам ЭВМ. Такое подключение адаптера обеспечивает достаточно быстрый обмен информацией между ЭВМ, при этом обмен может производиться большими массивами информации. В отношении скорости передачи информации связь через АКК мало уступает связи через общее ОЗУ, а в отношении объема передаваемой информации — связи через общее ВЗУ. Функции АКК достаточно просты: это устройство должно обеспечивать взаимную синхронизацию работы двух ЭВМ и буферизацию информации при ее передаче. Хотя функции АКК и его структура достаточно просты, однако большое разнообразие режимов работы двух ЭВМ и необходимость реализации этих режимов существенно усложняет это устройство.

Прямосвязанные комплексы позволяют осуществлять все способы организации ММВК, характерные для слабосвязанных комплексов. Однако за счет некоторого усложнения связей эффективность комплексов может быть значительно повышена. В частности, в прямосвязанных комплексах возможен быстрый переход от основной ЭВМ к резервной и в тех случаях, когда резервная ЭВМ загружена собственными задачами. Это позволяет обеспечивать высокую надежность при высокой производительности.

В реальных комплексах одновременно используется не один вид связи между ЭВМ, а два или более. В том числе очень часто в прямосвязанных комплексах присутствует и косвенная связь через ВЗУ.

Для комплексов с *спутниковыми ЭВМ* характерным является не способ связи, а принципы взаимодействия ЭВМ. Структура связей в спутниковых комплексах не отличается от связей в обычных ММВК: чаще всего связь между ЭВМ осуществляется через АКК. Особенностью же этих комплексов является то, что в них, во-первых, ЭВМ существенно различаются по своим характеристикам, а во-вторых, имеет место определенная соподчиненность машин и различие функций, выполняемых каждой ЭВМ. Одна из ЭВМ, основная, является, как правило, высокопроизводительной и предназначается для

основной обработки информации. Вторая, существенно меньшая по производительности, называется сателлитной или вспомогательной ЭВМ. Ее назначение — организация обмена информацией основной ЭВМ с периферийными устройствами, ВЗУ, удаленными абонентами, подключенными через аппаратуру передачи данных к основной ЭВМ. Кроме того, сателлитная ЭВМ может производить предварительную сортировку информации, преобразование ее в форму, удобную для обработки на основной ЭВМ, приведение выходной информации к виду, удобному для пользователя, и др. Сателлитная ЭВМ, таким образом, избавляет основную высокопроизводительную ЭВМ от выполнения многочисленных действий, которые не требуют ни большой разрядности, ни сложных операций, т. е. операций, для которых большая, мощная ЭВМ не нужна. Более того, с учетом характера выполняемых сателлитной машиной операций она может быть ориентирована на выполнение именно такого класса операций и обеспечивать даже большую производительность, чем основная ЭВМ.

Некоторые комплексы включают в себя не одну, а несколько сателлитных ЭВМ, при этом каждая из них ориентируется на выполнение определенных функций: например, одна осуществляет связь основной ЭВМ с устройствами ввода—вывода информации, другая — связь с удаленными абонентами, третья организует файловую систему и т. д.

Появление в последнее время дешевых и простых микро-ЭВМ в немалой степени способствует развитию сателлитных комплексов. Сателлитные комплексы решают только одну задачу: увеличивают производительность комплекса, не оказывая заметного влияния на показатели надежности.

Подключение сателлитных ЭВМ принципиально возможно не только через АКК, но и другими способами, однако связь через АКК наиболее удобна.

- **Мультипроцессорные компьютерные системы (МКС). Типы структурной организации и основы проектирования МКС**

Многопроцессорный вычислительный комплекс (МПВК) – это комплекс, включающий в себя два или более процессоров, имеющих общую оперативную память, общие периферийные устройства и работающих под управлением единой операционной системы (ОС), которая, в свою очередь, осуществляет общее управление техническими и программными средствами комплекса. Следует оговорить, что каждый из процессоров может иметь индивидуальные, доступные только ему ОЗУ и периферийные устройства. Все перечисленное весьма существенно, так как делает возможной гибкую организацию параллельной обработки информации и позволяет наиболее эффективно использовать все ресурсы комплекса.

На рис. 3 представлена упрощенная схема МПВК, содержащая три процессора, два модуля ОЗУ и одну подсистему ввода – вывода информации (ПВВ). Даже для такого простого варианта схема оказывается достаточно сложной, так как в МПВК должен быть обеспечен доступ любого процессора и любого канала ввода – вывода к любой ячейке ОЗУ, любого процессора к

любому каналу и периферийному устройству. Если представить теперь, что процессоров существенно больше, что ОЗУ по соображениям надежности и удобства наращивания емкости выполнено в виде нескольких модулей, а подсистема ввода – вывода включает в себе несколько каналов и большое число периферийных устройств, то становится ясным, насколько сложна топология МПВК. Если же учесть то обстоятельство, что для работы ОС аппаратные средства должны обеспечивать работу с переменными логическими адресами ОЗУ, периферийных устройств и каналов ввода – вывода, защиту памяти от взаимного влияния различных программ и возможность запуска одним процессором другого, сложность аппаратной реализации МПВК становится ясной в полной мере.

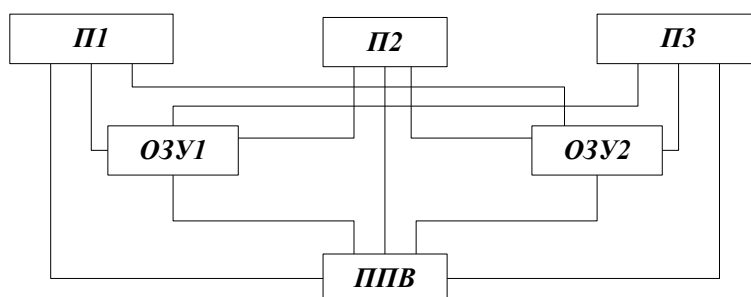


Рисунок 3 - Связи в МПВК

Непростые задачи возникают и при организации вычислительного процесса в МПВК., т. е. при построении ОС, которые являются основным средством организации всех процессов обработки информации в комплексе. Кроме обычных функций, выполняемых ОС при мультипрограммной обработке информации, возникают такие задачи, как распределение ресурсов и заданий между процессорами, синхронизация процессов при решении несколькими процессорами одной задачи, планирование с учетом оптимизации загрузки всех процессоров. При этом надо иметь в виду, что в процессе работы в комплексе возникает большое число конфликтных ситуаций, которые должны обрабатываться ОС. Эти и ряд других обстоятельств и факторов, связанных с обеспечением высокой надежности, делают ОС МПВК чрезвычайно сложной.

Однако, несмотря на все трудности, связанные с аппаратной и программной реализацией, МПВК получают все большее распространение, так как обладают рядом достоинств, основные из которых:

- высокая надежность и готовность за счет резервирования и возможности реконфигурации;
- высокая производительность за счет возможности гибкой организации параллельной обработки информации и более полной загрузки всего оборудования;
- высокая экономическая эффективность за счет повышения коэффициента использования оборудования комплекса.

Рассматривая процесс появления и развития МПВК, по-видимому, следует признать, что первоначально перед МПВК ставилась только задача обеспечения высокой надежности системы.

Неслучайно поэтому, что одним из первых (а может быть, и самым первым) МПВК был комплекс D-825, созданный фирмой «Барроуз» (США) в 1968 г. для систем военного назначения. Комплекс включал в себя 4 процессора, 16 модулей ОЗУ, 10 каналов ввода – вывода и до 256 периферийных устройств, т. е. был весьма представительным МПВК, по тем меркам. Надо сказать, что эта первая попытка была весьма удачной – поставленные задачи были полностью решены. Вместе с тем создание первых же МПВК выявили и возможности достижения с их помощью высокой производительности. В настоящее время МПВК чаще создаются именно с такой целью.

Типы структурной организации МПВК. Существует три типа структурной организации МПВК: с общей шиной; с перекрестной коммутацией; с многовходовыми ОЗУ.

В комплексах с *общей шиной* проблема связей всех устройств между собой решается крайне просто: все они соединяются общей шиной, выполненной в виде совокупности проводов или кабелей, по которым передаются информация, адреса и сигналы управления (рис. 4, а). Интерфейс является односвязным, т. е. обмен информацией в любой момент времени может происходить только между двумя устройствами. Если потребность в обмене существует более чем у двух устройств, то возникает конфликтная ситуация, которая разрешается с помощью системы приоритетов и организации очередей в соответствии с этим. Обычно функции арбитра выполняет либо процессор, либо специальное устройство, которое регистрирует все обращения к общей шине и распределяет шину во времени между всеми устройствами комплекса.

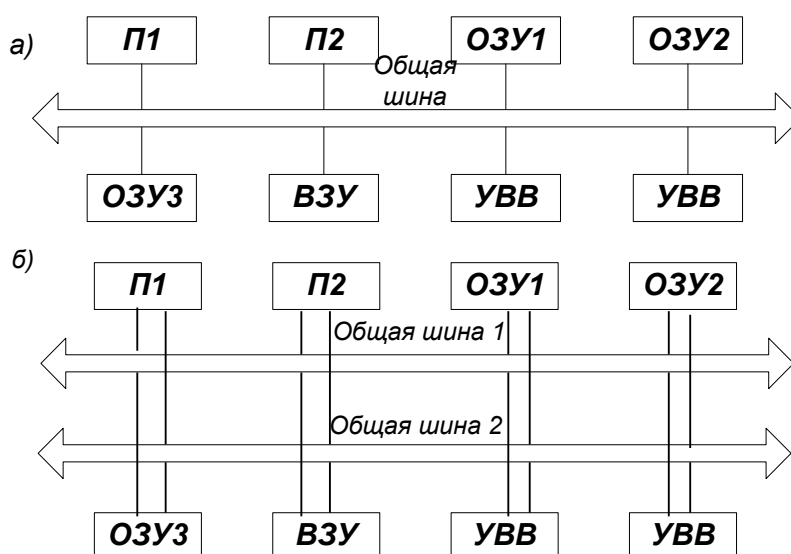


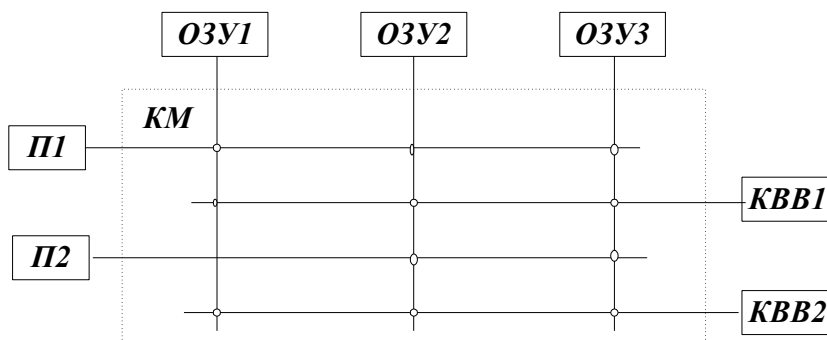
Рисунок 4 - МПВК с общей шиной

Несомненные достоинства структуры с общей шиной – простота, в том числе изменения комплекса, добавления или изъятия отдельных устройств, а также доступность модулей ОЗУ для всех остальных устройств. Следствием всего этого является достаточно низкая стоимость комплекса.

Вместе с тем комплексы с общей шиной не лишены определенных недостатков. Первый – невысокое быстродействие, так как одновременный обмен информацией возможен между двумя устройствами, не более. По этой причине в комплексах с общей шиной число процессоров не превосходит двух-четырёх. Этот недостаток может быть несколько компенсирован путем использования общей шины с высоким быстродействием, большим, чем быстродействие входящих в комплекс устройств. Однако этот путь приводит к усложнению и удорожанию комплекса. Вторым недостатком МПВК с общей шиной является в относительно низкой надежности системы из-за наличия общего элемента – шины. Надо иметь в виду, что надежность общей шины определяется не только надежностью проводов и кабелей (их собственная надежность достаточно высока), но и надежностью всех соединений, входных и выходных цепей устройства. Отказ хотя бы одного из элементов приводит к отказу всего комплекса. Этот недостаток можно компенсировать за счет введения резервной шины (рис. 4, б). Хотя это несколько усложняет комплекс, однако надежность его существенно возрастает. Если же резервную шину сделать активной, т. е. работающей одновременно с основной, то можно не только повысить надежность, но и увеличить производительность комплекса за счет того, что обмен информацией может осуществляться одновременно между двумя парами устройств.

Общая шина может быть организована различными способами – принципиально так же, как и для однопроцессорных ЭВМ с общей шиной. Характерным примером является комплекс СМ-1420, в котором используется общая шина однопроцессорных ЭВМ этой системы.

Полностью лишены недостатков, присущих МПВК. с общей шиной, МПВК с перекрестной коммутацией. Идея структурной организации таких ВК заключается в том, что все связи между устройствами осуществляются с помощью специального устройства – коммутационной матрицы (рис. 5, а). Коммутационная матрица (КМ) позволяет связывать друг с другом любую пару устройств, причем таких пар может быть сколько угодно: связи не зависят друг от друга.



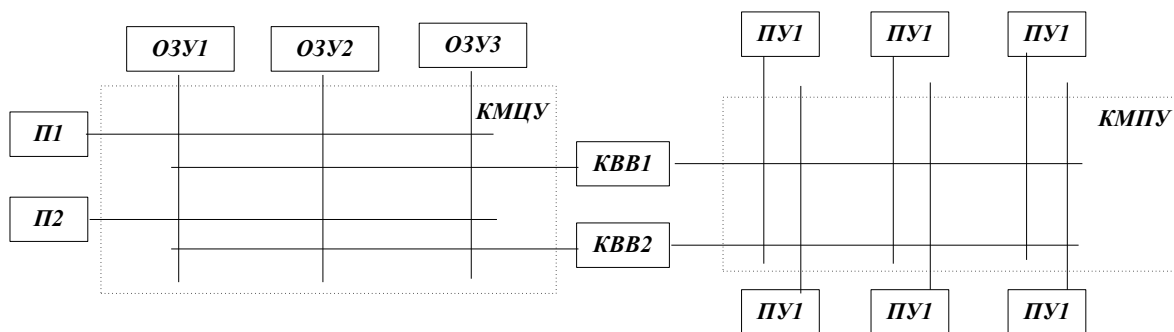


Рисунок 5 - МПВК с перекрестной коммутацией

В МПВК с перекрестной коммутацией нет конфликтов из-за связей, остаются только конфликты из-за ресурсов. Возможность одновременной связи нескольких пар устройств позволяет добиваться очень высокой производительности комплекса. Важно отметить и такое обстоятельство, как возможность установления связи между устройствами на любое, даже на длительное время, так как это совершенно не мешает работе других устройств, зато позволяет передавать любые массивы информации с высокой скоростью, что также способствует повышению производительности комплекса. Заметим, что в МПВК с общей шиной передача информации массивами, т. е. занятие типов одной парой устройств на длительный отрезок времени, обычно допускается лишь в крайних случаях, так как это приводит к длительным простоям остальных устройств.

Кроме того, к достоинствам структуры с перекрестной коммутацией можно отнести простоту и унифицированность интерфейсов всех устройств, а также возможность разрешения всех конфликтов в коммутационной матрице. Важно отметить и то, что нарушение какой-то связи приводит не к выходу из строя всего комплекса, а лишь к отключению какого-либо устройства, т. е. надежность таких комплексов достаточно высока. Однако и организация МПВК с перекрестной коммутацией не свободна от недостатков.

Прежде всего – сложность наращивания ВК. Если в коммутационной матрице заранее не предусмотреть большого числа входов, то введение дополнительных устройств в комплекс потребует установки новой коммутационной матрицы. Существенным недостатком является и то, что коммутационная матрица при большом числе устройств в комплексе становится сложной, громоздкой и достаточно дорогостоящей. (Надо учитывать то обстоятельство, что коммутационные матрицы строятся обычно на схемах, быстродействие которых существенно выше быстродействия схем и элементов основных устройств, – только при этом условии реализуются все преимущества коммутационной матрицы.) Это обстоятельство в значительной степени усложняет и удорожает комплексы.

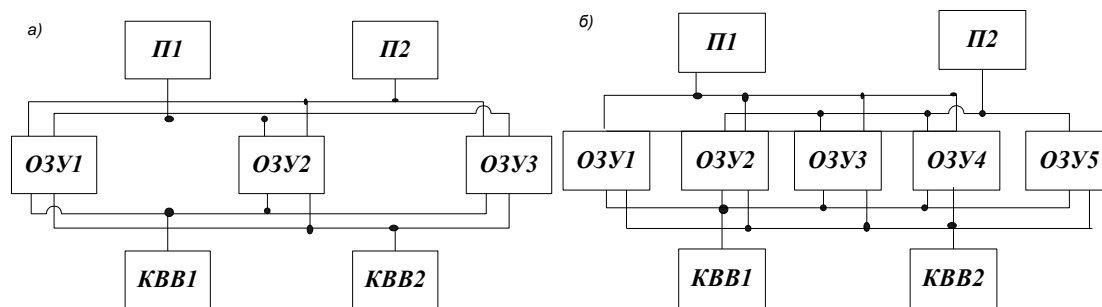


Рисунок 6 - МПВК с многовходовым ОЗУ

Для того чтобы упростить и удешевить ВК, коммутацию устройств осуществляют с помощью двух и даже более коммутационных матриц. На рис. 2.8, б представлен МПВК, включающий в себя две матрицы: КМЦУ – матрицу для центральных устройств (процессоров, ОЗУ и каналов ввода – вывода) и КМПУ – матрицу для периферийных устройств. Схемы последней могут иметь существенно меньшее быстродействие, чем схемы первой, да к тому же обе коммутационные матрицы будут значительно проще и дешевле, чем одна общая коммутационная матрица с высоким быстродействием.

Перекрестная коммутация довольно широко используется при построении ВК, в частности практически всех МПВК фирмы «Барроуз» (в том числе и упомянутого выше комплекса D-825).

В МПВК с многовходовыми ОЗУ все, что связано с коммутацией устройств, осуществляется в ОЗУ. В этом случае модули ОЗУ имеют число входов, равное числу устройств, которые к ним подключаются, т. е. для каждого устройства предусматривается свой вход в ОЗУ. Структура такого МПВК показана на рис. 6, а. В отличие от ВК с перекрестной коммутацией, которые имеют централизованное коммутационное устройство, в МПВК с многовходовыми ОЗУ средства коммутации распределены между несколькими устройствами. Такой способ организации МПВК сохраняет все преимущества систем с перекрестной коммутацией, несколько упрощая при этом саму систему коммутации. Для наращивания системы должны быть предусмотрены дополнительные входы в ОЗУ. Правда, введение дополнительных модулей ОЗУ не вызывает затруднений.

В МПВК с многовходовыми ОЗУ очень просто решается вопрос о выделении каждому процессору своей оперативной памяти, недоступной другим процессорам. Такая организация показана на рис. 6, б. Выделение индивидуальной памяти каждому процессору позволяет хранить в ней информацию, которая необходима только одному процессору: различные таблицы и данные, копии некоторых модулей операционной системы и др. Это позволяет избежать части конфликтов, которые неизбежно возникают при общей оперативной памяти. Кроме того, уменьшается вероятность искажения информации в ОЗУ другими процессорами. Однако такие БК имеют тот недостаток, что в случае выхода из строя какого-либо процессора доступ к его памяти затруднен и информация может быть переписана в другой модуль ОЗУ только через канал ввода – вывода и внешнее ЗУ, что требует много времени.

Приведенные три типа структурной организации исчерпывают существующие построения МПВК, полностью удовлетворяющих тому определению, которое дано в начале этого параграфа. Такие МПВК в литературе называют классическими или истинными МПВК.

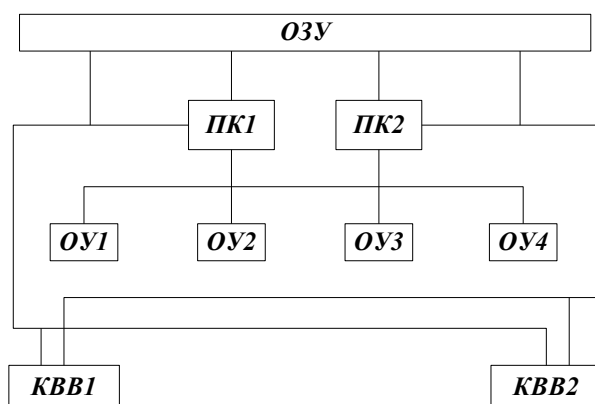


Рисунок 7 - Комплекс ЕС-1065.

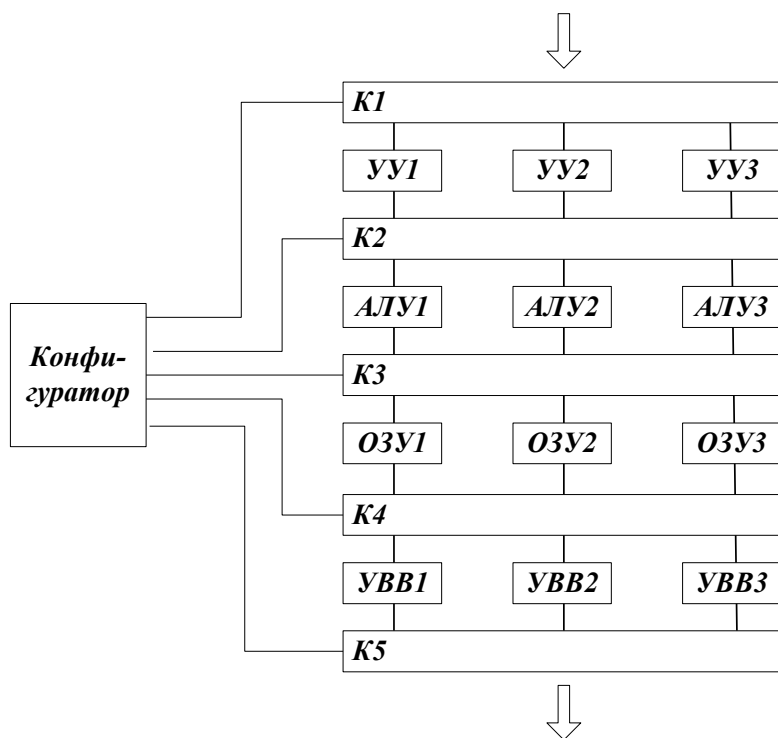


Рисунок 8 - Комплекс повышенной надежности

Вместе с тем нередко МПВК называют комплексы, лишь частично удовлетворяющие этому определению, например любые комплексы, в которых имеется несколько процессоров, а иногда даже матричные и конвейерные, которые (в отличие от классических МПВК) не относятся к комплексам с множественным потоком команд и множественным потоком данных. Кроме того, иногда к МПВК относят ЭВМ, имеющие не одно общее арифметическо-логическое устройство, а несколько операционных устройств (ОУ), каждое из которых выполняет определенную группу операций и может работать

совершенно независимо от других. Таким образом, на первый взгляд может показаться, что такой комплекс имеет несколько процессоров, каждый из которых связан с общим ОЗУ. Однако все ОУ работают под управлением одной программы.

Но вот ЭВМ ЕС-1065, в которой применяются множественные ОУ, с полным правом можно называть МПВК, так как кроме этих ОУ имеется два процессора команд (ПК), причем каждый может работать по собственной программе (рис. 7). В этом комплексе налицо все элементы: несколько независимых процессоров, работающих с общедоступными ОЗУ, имеющих общие периферийные устройства и работающих под управлением общей операционной системы. По типу структурной организации комплекс ЕС-1065 может быть отнесен к МПВК с многовходовыми ОЗУ.

Нередко к МПВК относят комплексы, являющиеся по существу ММВК, но в которых для достижения более высокой надежности резервирование осуществляется не «помашинно», а по устройствам. Структурная схема одного из таких ВК приведена на рис. 8. Комплекс первоначально работает как трехмашинный со сравнением результатов на выходе. Результат используется по методу «голосования». При несовпадении результата с помощью системы контроля определяется и отключается неисправное устройство. Эти функции выполняются конфигуратором, контролирующим состояние комплекса. Такой комплекс обладает высокой надежностью, однако, как видим, несмотря на наличие трех процессоров, его нельзя назвать многопроцессорным.

Контрольные вопросы:

1. В чем заключается отличие сосредоточенных СОД от распределенных?
2. Поясните классификацию вычислительных систем, комплексов.
3. Основные характеристики ВС и ВК.
4. Способы оценки эффективности ВС, ВК и сетей.
5. Какие существуют виды связей в ММВС?
6. Мультипроцессорные компьютерные системы.
7. Типы структурной организации МПВК.
8. В чем заключается задача распараллеливания процессов в ЭВМ?
9. Какие существуют модели обмена сообщениями?
10. Протоколы управления сетью.

Тема 12. Основы архитектуры персональных компьютеров

1. Этапы развития процессоров современных ЭВМ.
2. Архитектура процессоров современных персональных компьютеров (ПК).
3. Модели процессоров. Качественные отличия процессоров различных фирм.

1 Этапы развития процессоров современных ЭВМ

Первые микропроцессоры (МП) появились в начале 70-х годов в результате совместных усилий системотехников, решающих проблемы архитектурной организации средств вычислительной техники, и схемотехников, занимающихся вопросами конструирования и технологии производства радиоэлектронных средств. Первый МП поступил на неподготовленный к этому событию рынок в 1971 г. и предстал перед миром как знамение новой эры интегральной электроники. Рассмотрим эволюцию МП.

Семейство процессоров x88/x86.

Первый микропроцессор – 4-х разрядный i4004 - был изготовлен 1971 году и с тех пор фирма Intel (INTEgrated Electronics) прочно удерживает лидирующее положение на данном сегменте рынка. Стоит, пожалуй, напомнить, что максимальная тактовая чистота этого прапрадедушки современных «числодробилок» составляла всего 750 кГц.

Реализация ряда следующих проектов фирмы Intel по разработке однокристальных микропроцессоров (i4040, i8008) возвестила о наступлении новой эры персональных компьютеров. Наиболее успешным был, пожалуй, проект разработки микропроцессора i8080. Кстати, впоследствии именно на этом микропроцессоре был основан компьютер «Альтаир», для которого молодой Бил Гейтс написал свой первый интерпретатор Бейсика. Этот 8-разрядный микропроцессор был выполнен по n-канальной МОП-технологии (n-MOS), а его тактовая частота не превышала 2 МГц. Не будет преувеличением сказать, что классическая архитектура i8080 оказала огромное влияние на дальнейшее развитие однокристальных микропроцессоров. Несмотря на заслуженный успех i8080, настоящим промышленным стандартом для персональных компьютеров стал другой микропроцессор фирмы Intel.

i8088

Микропроцессор i8088 был анонсирован Intel в июне 1979 года, а в 1981-м фирма IBM выбрал этот микропроцессор для своего первого персонального компьютера и, надо сказать не ошибся. Новый чип содержал примерно 29 тысяч транзисторов. Одним из существенных достоинств микропроцессора i8088 была возможность (благодаря 20 адресным линиям) физически адресовать область памяти в 1 Мбайт. Здесь следует, правда, отметить, что для IBM PC в этом пространстве было отведено всего лишь 640 Кбайт. Хотя с внешними периферийными устройствами (дисками, видео) i8088 был связан внешней 8-разрядную шину данных, его внутренняя структура (адресуемые регистры) позволяла работать с 16-разрядными словами. Первоначально

микропроцессор i8088 работал на частоте 4,77 МГц и имел быстродействие 0,33 MIPS (Million Instruction Per Second), однако впоследствии были разработаны его клоны, рассчитанные на более высокую тактовую частоту (например, 8 МГц).

i8086

Чип 8086, появившийся ровно на год раньше своего счастливого последователя (в июне 1978 года), стал популярен благодаря компьютеру Compaq DeskPro. Программная модель (доступные регистры) этого микропроцессора полностью совпадает с моделью i8088. Основное отличие данных микропроцессора состоит в различной разрядности внешней шины данных: 8 разрядов у i8088 и 16 у i8086. Понятно, что более высокой производительности с новым микропроцессором можно было достичь при использовании компьютера, на системной шине которого под данные предусмотрено 16 линий. Адресная шина микропроцессора i8086 по-прежнему позволяла адресовать 1 Мбайт памяти.

i80286

Опираясь на архитектуру i8086 и учитывая запросы рынка, в феврале 1982 года фирма Intel выпустила свой новый микропроцессор - i80286. На кристалле было реализовано около 130 тысяч транзисторов. Надо сказать, что этот чип появился практически одновременно с новым компьютером фирмы IBM - PC/AT. Наряду с увеличением производительности этот микропроцессор (i80286) мог теперь работать в двух режимах - реальном и защищённом. Если первый режим был (за рядом исключений) похож на обычный режим работы i8088/86, то второй использовал более изощрённую технику управления памятью. В частности, защищённый режим работы позволял, например, таким программным продуктам, как Windows 3.0 и OS/2, работать с оперативной памятью свыше 1 Мбайта. Благодаря 16 разрядам данных на новой системной шине, которая была впервые использована в IBM PC/AT286, мог обмениваться с периферийными устройствами 2-байтными сообщениями. 24 адресные линии нового микропроцессора позволяли в защищённом режиме обращаться уже к 16 Мбайтам памяти. В микропроцессоре i80286 впервые на уровне микросхем были реализованы многозадачность и управление виртуальной памятью. При тактовой частоте 8 МГц достигалась производительность 1,2 MIPS.

i80386DX

В октябре 1985 года фирмой Intel был анонсирован (представлен) первый 32-разрядный МП i80386. Новый чип содержал примерно 275 тысяч транзисторов. Первым компьютером, использующий этот процессор, был Compaq Desk Pro 386 (другие источники говорят о лидерстве фирмы ALR). Полностью 32-разрядная архитектура (32-разрядные регистры и 32-разрядная внешняя шина данных) в новом МП была дополнена расширенным устройством управления памятью MMU (Memory Management Unit), которая помимо блока сегментации (Segmentation Unit) было дополнено блоками управления страницами (Paging Unit). На тактовой частоте 16 МГц быстродействие нового процессора составляло примерно 6 MIPS. 32 адресные линии микропроцессора позволяли физически адресовать 4 Гбайта памяти.

Кроме того, был введён новый режим - виртуального процессора (V86). В этом режиме могли одновременно выполняться несколько задач, предназначенных для i8086.

i80386SX

Более дешёвая альтернатива 32-разрядному процессору i80386, который в последствии получил окончание DX, появился только в июне 1988 года. То был процессор i80386SX. В отличие от своего старшего « собрата », новый МП использовал 16-разрядную шину внешних данных и 24-разрядную адресную (адресуемое пространство - 16 Мбайт). Это было особенно удобно для стандарта PC/AT, системная шина которых использует, как известно, только 16 линий данных. Благодаря дешевизне нового изделия многие производители «железа» стали заменять уже устаревший микропроцессор i80286 на более производительный i80386SX. Одним из решающих факторов для замены была полная совместимость 32-разрядных микропроцессоров: программное обеспечение, написанное для i80386DX, корректно работало и на i80386SX. Дело в том, что внутренние регистры их были полностью идентичны. Индекс SX произошёл от слова SiXteen (шестнадцать), поскольку разрядность внешней шины данных нового тогда процессора была именно такой. В дальнейшем, правда, для 486-х процессоров SX стал означать отсутствие математического сопроцессора.

i486

На осенней выставке Comdex в 1989 году фирма Intel впервые анонсировала МП i486DX, который содержал более миллиона транзисторов (а точнее, 1,2 миллиона) на одном кристалле и был полностью совместим с процессором ряда x86. Напомним, что на кристалле первого члена этого семейства - микропроцессора i8088 - насчитывалось около 29 тысяч транзисторов. В борьбе с микропроцессорами-клонами фирма Intel намеренно убрала из названия нового устройства число 80. Новая микросхема впервые объединила на одном чипе такие устройства, как центральный процессор, математический сопроцессор и кэш-память. Использование конвейерной архитектуры, присущей RISC-процессорам, позволило достичь четырёхкратной производительности обычных 32-разрядных систем. Это связано с уменьшением количества тактов для реализации каждой команды. 8-Кбайтная встроенная кэш-память ускоряет выполнение программ за счёт промежуточного хранения часто используемых команд и данных. На тактовой частоте 25 МГц микропроцессор показал производительность 16,5 MIPS. Созданная в июне 1991 года версия микропроцессора с тактовой частотой 50 МГц позволила увеличить производительность ещё на 50%. Встроенный математический сопроцессор существенно облегчал и ускорял математические вычисления.

i486SX

Появление нового микропроцессора i486SX фирмы Intel вполне можно считать одним из важнейших событий 1991 года. Уже предварительные испытания показали, что компьютеры на базе i486SX с тактовой частотой 20 МГц работают быстрее (примерно на 40%) компьютеров, основанных на

i80386DX с тактовой частотой 33 МГц. Микропроцессор i486SX, подобно оригинальному i486DX, содержит на кристалле и кэш-память, а вот математический сопроцессор у него заблокирован. Значительная экономия (благодаря исключению затрат на тестирование сопроцессора) позволила фирме Intel существенно снизить цены на новый микропроцессор. Надо сказать, что если микропроцессор i486DX был ориентирован на применение в сетевых серверах и рабочих станциях, то i486SX послужил отправной точкой для создания мощных настольных компьютеров. Вообще говоря, в семействе микропроцессоров i486 предусматривается несколько новых возможностей для построения мультимикропроцессорных систем: соответствующие команды поддерживают механизм семафоров памяти, аппаратно-реализованное выявление недостоверности строки кэш-памяти обеспечивает согласованность между несколькими модулями кэш-памяти и т.д. Для микропроцессоров семейства i486 допускается адресация физической памяти размером 64 Тбайт.

i80386SL

К концу 1991 года 32-разрядные МП стали стандартными для компьютеров типа лэптоп и ноутбук, однако обычные микросхемы i80386DX/SX не полностью отвечали требованиям разработчиков портативных компьютеров. Для удовлетворения потребностей этого сегмента рынка в 1990 году фирмой Intel был разработан микропроцессор i80386SL, который содержал примерно 855 тысяч транзисторов. Данный микропроцессор представляет собой интегрированный вариант микропроцессора i80386SX, базовая архитектура которого дополнена ещё несколькими вспомогательными контролерами. По существу, все компоненты, необходимые для построения портативного компьютера, сосредоточены в двух микросхемах: микропроцессоре i80386SL и периферийном контролере i82360SL.

i486SL

Микросхема i486SL представляет собой самый производительный процессор серии SL, разработанный фирмой Intel. Анонсированная в конце 1992 года, эта микросхема объединяет характерные черты двух представителей процессорных семейств Intel: i486DX и i80386SL. По производительности новый процессор не уступает i486DX, но благодаря пониженному напряжению питания (3,3 В) и развитой технологии управления энергоснабжением (как в i80386SL) он может эффективно использоваться в портативных компьютерах. По некоторым оценкам, системная плата компьютера на базе i486SL примерно на 60% меньше, чем при использовании i80386SL. С конца 1993 года фирма Intel начала выпускать новую серию микропроцессоров 486SL Enhanced, которая заменила 5-вольтовые 486SX, 486DX, 486DX2, и OverDrive-процессоры. Подобные процессоры имеют напряжение питания 3,3 В и развитую технологию энергосбережения, что полностью соответствует американской национальной программе Energy Star.

Процессоры с умножением частоты

В марте 1992 года фирма Intel объявила о создании второго поколения микропроцессоров 486. Эти микропроцессоры, названные i486DX2, обеспечили новую технологию, при которой скорость работы внутренних блоков

микропроцессора в два раза выше скорости остальной части системы. Тем самым появилась возможность объединения высокой производительности микропроцессора с внутренней тактовой частотой 50(66) МГц и эффективной по стоимости 25/33-мегагерцовой системной платой. Новые микропроцессоры по-прежнему включали в себя центральный процессор, математический сопроцессор и кэш-память на 8 Кбайт. Компьютеры, поставляемые на базе микропроцессоров i486DX2, работают приблизительно на 70% производительней тех, что основаны на микропроцессорах i486DX первого поколения. Несколько позже появились процессоры i486SX2, в которых, как следует из названия, отсутствует встроенный сопроцессор.

В настоящее время технология умноженной частоты (не только в два, но и, например, в полтора, два с половиной или три раза) находит широкой практическое применение во всех современных процессорах. Так, фирма Intel выпускала серию микропроцессоров с умножением частоты - DX4 (кодовое название P24C). Процессоры этого семейства - 486DX4-75, 486DX4-83 и 486DX4-100 имеют кэш-память 16 Кбайт и предназначены для установки с системные платы, работающие на тактовой частоте 25 и 33 МГц. Напряжение питания этих процессоров составляет 3,3 В, количество транзисторов на кристалле - 1,6 миллиона.

Pentium

В марте 1993 года фирма Intel объявила о начале промышленных поставок 66- и 60-мегагерцовых версий процессора Pentium, известного ранее как 586 или P5. Название нового микропроцессора является зарегистрированной торговой маркой корпорации Intel . Таким образом, в системах Intel Inside микропроцессор 586 фигурировать не будет. Системы, построенные на базе Pentium, полностью совместимы со 100 миллионов персональных компьютеров, использующих микропроцессоры i8088, i80286, i80386, i486. Новая микросхема содержит около 3,1 миллиона транзисторов и имеет 32-разрядную адресную и 64-разрядную внешнюю шину данных, что обеспечивает обмен данными с системной платой со скоростью до 528 Мбайт/с. В отличие от 486-х процессоров, для производства которых использовалась CMOS-технология, для Pentium фирмы Intel применила 0,8-микронную BiCMOS-технологиию.

Pentium с тактовой частотой 66 МГц имеет производительность 112 MIPS (миллионов операций в секунду). В настоящее время микросхемы Pentium сняты с производства.

Pentium Pro

1 ноября 1995 года фирма Intel объявила о начале коммерческих поставок микропроцессора нового поколения Pentium Pro именуемого до недавнего времени P6. В его основе лежит комбинация технологий, известная как Dynamic Execution. Собственно, это три уже известные технологии: многократное предсказание ветвлений, анализ потоков данных и эмуляция выполнения инструкций. В корпусе микросхемы размещены два кристалла, одним из которых является 256- или 512-Кбайтная кэш-память второго уровня. На кристалле процессора, как обычно, расположен 16-Кбайтный кэш. На сегодняшний день в семейство Pentium Pro входят микропроцессоры с тактовой

частотами 200, 180, 166 и 150 МГц. Архитектура Pentium Pro позволяет соединять между собой множество процессоров, создавая, таким образом, непревзойдённую масштабируемость. Так, Министерство энергетики США создала систему, базирующуюся на 9 тысячах процессоров.

Pentium II Xeon

С начала июля 1998 года по всему миру проходила серия мероприятий, посвящённых представлению самого мощного процессора архитектуры x86 корпорации Intel. Первое, что бросается в глаза, - необычно крупный размер процессорного картриджа в который «пакуется» Xeon. Он предназначен для установки в разъём новой конструкции Slot 2. По словам разработчиков, это связано с увеличением ёмкости кэш-памяти второго уровня. В настоящий момент процессоры Xeon с единой тактовой частотой поставляются в двух вариантах: с 512 Кбайт и 1 Мбайт кэша L2. Планируется довести ёмкость кэш-памяти второго уровня до 2 Мбайт и повысить тактовую частоту до 450 МГц. Напомню, что старый Pentium II комплектовался лишь 512 Кбайт.

Но ещё больший интерес вызывает тот факт, что конструкторы смогли «заставить» L2-кэш работать на тактовой частоте процессорного ядра. Та же концепция была реализована в Pentium Pro, но при этом разработчики «столкнулись» на стадии производства (процент выхода двух качественных кристаллов оказался ниже предполагаемого), и процессор оказался довольно дорогим. Возможно, именно поэтому Pentium II изначально создавался с «разделением» кристаллов (основного и кэша L2), за что пришлось расплачиваться «половиной» тактовой частоты кэш-памяти второго уровня.

Высокая частота работы кэша спровоцировала увеличение теплоотдачи процессорного блока, поэтому потребовалось использование массивной поглощающей тепло пластины, что, в свою очередь, привело к увеличению веса и габаритов модуля.

Усовершенствованная архитектура Xeon, допускающая 36-разрядную адресацию физической памяти, теоретически позволяет процессору получать доступ к системной памяти ёмкостью до 64 Гбайт. Новый механизм постраничного обмена Page Size Extension - 36 останется практически незаметной для глаз пользователя и разработчиков приложений. В настоящее время PSE-36 поддерживают операционные системы Windows NT, SCO UnixWare и Sun Solaris. Для остальных операционных систем потребуется обновить драйвер блока управления памятью.

Xeon, предназначен не только для серверов, но и для рабочих и графических станций, для которых одним из важнейших параметров является производительность видеоподсистемы. Для них разработан чипсет Intel 440GX AGPset на базе известного микросхемного набора 440BX. 440GX управляет работой порта AGP в режиме 2x. Режим удвоенной производительности реализуется благодаря так называемой технологии двойной накачки - данные передаются как по переднему, так и по заднему фронтам тактовых импульсов (у обычной AGP - только по переднему), при этом полоса пропускания достигает

значения 533 Мбайт/с. Физические параметры интерфейса AGP остаются прежними.

Ещё одной особенностью набора чипсета 440GX стала возможность обращения к памяти ёмкостью до 2 Гбайт, что в два раза больше, чем у его приемника.

2 Архитектура процессоров современных персональных компьютеров (ПК)

Существует несколько подходов к классификации микропроцессоров по типу архитектуры. Так, выделяют МП с CISC (Complete Instruction Set Computer) архитектурой, характеризуемой полным набором команд, и RISC (Reduce Instruction Set Computer) архитектурой, которая определяет систему с сокращенным набором команд одинакового формата, выполняемых за один такт МП.

Определяя в качестве основной характеристики МП разрядность, выделяют следующие типы МП архитектуры:

- с фиксированной разрядностью и списком команд (однокристалльные);
- с наращиваемой разрядностью (секционные) и микропрограммным управлением.

Анализируя адресные пространства программ и данных, определяют МП с архитектурой фон Неймана (память программ и память данных находятся в едином пространстве и нет никаких признаков, указывающих на тип информации в ячейке памяти) и МП с архитектурой Гарвардской лаборатории (память программ и память данных разделены, имеют свои адресные пространства и способы доступа к ним).

Рассмотрим более подробно основные типы архитектурных решений, выделяя связь со способами адресации памяти.

1. Регистровая архитектура определяется наличием достаточно большого регистрового файла внутри МП. Команды получают возможность обратиться к операндам, расположенным в одной из двух запоминающих сред: оперативной памяти или регистрах. Размер регистра обычно фиксирован и совпадает с размером слова, физически реализованного в оперативной памяти. К любому регистру можно обратиться непосредственно, поскольку регистры представлены в виде массива запоминающих элементов - регистрового файла. Типичным является выполнение арифметических операций только в регистре, при этом команда содержит два операнда (оба операнда в регистре или один операнд в регистре, а второй в оперативной памяти).

К данному типу архитектуры относится микропроцессор фирмы Zilog. Процессор Z80 - детище фирмы Zilog помимо расширенной системы команд, одного номинала питания и способности исполнять программы, написанные для i8080, имел архитектурные "изюминки".

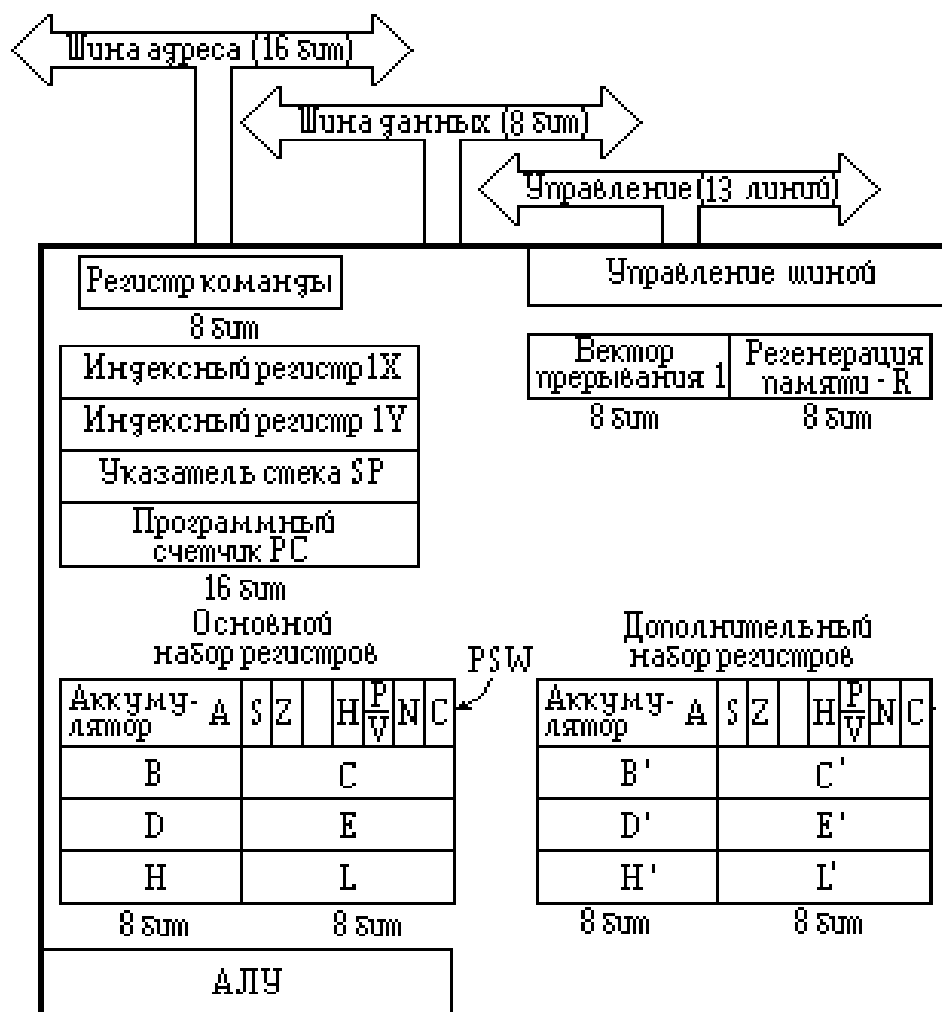


Рисунок 1 - Микропроцессор Z80 фирмы Zilog.

В дополнение к основному набору РОН, в кристалле был реализован второй комплект аналогичных регистров. Это значительно упрощало работу при вызове подпрограмм или процедур обслуживания прерываний, поскольку программист мог использовать для них альтернативный набор регистров, избегая сохранения в стеке содержимого РОНов для основной программы с помощью операций PUSH. Кроме того, в систему команд был включен ряд специальных инструкций, ориентированных на обработку отдельных битов, а для поддержки регенерации динамической памяти в схему процессора введены соответствующие аппаратные средства. Z80 применялся в машинах Sinclair ZX, Sinclair Spectrum, Tandy TRS80.

Предельный вариант - архитектура с адресацией посредством аккумуляторов (меньший набор команд).

МП фирмы Motorola имел ряд существенных преимуществ. Прежде всего, кристалл MC6800 требовал для работы одного номинала питания, а система команд оказалась весьма прозрачной для программиста. Архитектура МП также имела ряд особенностей.

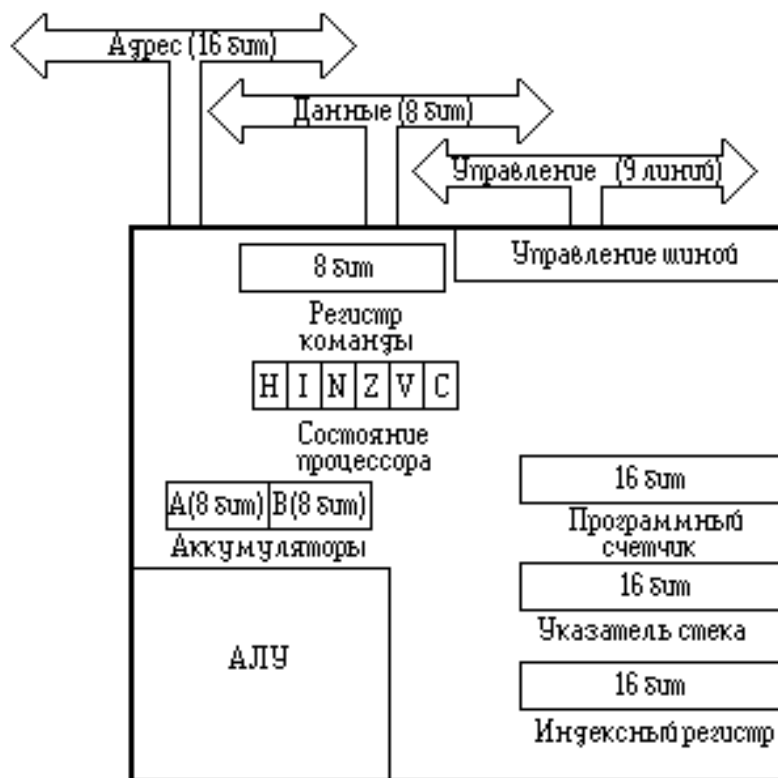


Рисунок 2 - Микропроцессор MC6800 фирмы Motorola

Микропроцессор MC6800 содержал два аккумулятора, и результат операции АЛУ мог быть помещен в любой из них. Но самым ценным качеством структуры MC 6800 было автоматическое сохранение в стеке содержимого всех регистров процессора при обработке прерываний (Z80 требовалось для этого несколько команд PUSH). Процедура восстановления РОН из стека тоже выполнялась аппаратно.

2. Стековая архитектура дает возможность создать поле памяти с упорядоченной последовательностью записи и выборки информации.

В общем случае команды неявно адресуются к элементу стека, расположенному на его вершине, или к двум верхним элементам стека.

3. Архитектура МП, ориентированная на оперативную память (типа "память-память"), обеспечивает высокую скорость работы и большую информационную емкость рабочих регистров и стека при их организации в оперативной памяти.

Архитектура этого типа не предполагает явного определения аккумулятора, регистров общего назначения или стека; все операнды команд адресуются к области основной памяти.

С точки зрения важности для пользователя-программиста под архитектурой в общем случае понимают совокупность следующих компонентов и характеристик:

- разрядности адресов и данных;
- состава, имен и назначения программно-доступных регистров;
- форматов и системы команд;
- режимов адресации памяти;

- способов машинного представления данных разного типа;
- структуры адресного пространства;
- способа адресации внешних устройств и средств выполнения операций ввода/вывода;
- классов прерываний, особенностей инициирования и обработки прерываний.

3 Модели процессоров. Качественные отличия процессоров различных фирм

Процессоры фирм AMD, IBM, Cyrix и Texas Instruments.

Фирма AMD традиционно выпускает процессоры, совместимые с передовыми моделями от Intel. Эти процессоры обычно появляются несколько позже, но вбирают в себя достижения, реализованные Intel в более поздних моделях. Процессоры класса 486 фирмы AMD совместимы с моделями Intel. Наибольший интерес представляют процессоры семейства Enhanced Am486® и Am5X86™, представляющие вершину достижений, реализованных в рамках шины 486 процессора (Pentium OverDrive, конечно, их несколько превосходит, но его цена менее привлекательна). Их отличие экономичность потребления – питание пониженным напряжением, наличие развитых средств SMM и управления потреблением, более широкое применение политики обратной записи первичного кэша.

Процессоры используют умножение частоты на коэффициент 2,3 и даже 4, который может снижаться заземлением вывода CLKMUL.

Процессоры имеют возможность снижения энергопотребления в нерабочем режиме (аналогичные средства появились в процессорах Pentium начиная только со 2-го поколения). По сигналу STOPCLK# процессор выгружает буферы записи и входит в режим Stop Grant, в котором прекращается тактирование большинства узлов процессора, что вызывает снижение потребления. В этом состоянии он прекращает исполнение инструкций и не обслуживает прерывания, но продолжает слежение за шиной данных, отслеживание кэш-попадания. Из этого состояния процессор выходит по снятию сигнала STOPCLK#, совместно с использованием режима SMM, реализует механизм расширенного управления питанием APM (Advanced Power Management).

В состоянии пониженного потребления Auto HALT PowerDown процессор переходит при исполнении инструкции HALT. В этом состоянии процессор реагирует на все прерывания и также продолжает слежение за шиной.

Из состояния Stop Grant остановкой внешней синхронизации процессор можно перевести в режим Stop Clock, в котором он потребляет минимальную мощность. В этом режиме он не выполняет никаких функций, но при возобновлении синхронизации вернется в состояние Stop Grant, из которого можно выйти в нормальный режим работы.

Расширенные средства SMM, реализованные в процессоре, поддерживают рестарт инструкций ввода/вывода и изменение базового адреса SMRAM.

Процессоры Enhanced Am486 имеют обозначения вида

A80486 DX4 – 120 ля названия (слева направо) расшифровываются следующим образом:

- Типа корпуса: A=PGA-186, S=SQFP-208.
- Типа устройства: 80486 Am486.
- Версия: DX4 = с удвоением частоты и FPU, DX2 = с удвоением частоты и FPU.
- Частота (внутренняя), МГц: 120, 100, 80, 75 или 66.
- Семейство: S = ENHANCED (с расширенными возможностями).
- Напряжение питания: V = питание 3,3 В, входы допускают уровень сигнала 5 В.
- Размер кэша: 8 = 8 Кбайт.
- Тип кэша: B = Write Back.

Эти процессоры могут устанавливаться практически в любые системные платы с сокетом 1, 2 или 3, имеющими регулятор напряжения питания процессора, обеспечивающий номинальное напряжение 3,3 В. Платы, не поддерживающие расширенный режим шины, будут использовать процессоры только в режиме сквозной записи кэша. Более современные платы реализуют все преимущества данных процессоров.

Процессоры Am5x86-P75, они же AMD-X5-133 – самые высокопроизводительные процессоры класса 486 – имеют иную систему обозначения. Здесь надпись вида AMD-X5 – 133 A D W расшифровывается следующим образом:

- AMD-X5 – обозначение процессора с учетверением частоты.
- Частота (внутренняя) - 133 МГц.
- Тип корпуса: A=PGA-168, S=SQFP-208.
- Напряжение питания: D = 3,45 В, F = 3,3 В.
- Допустимая температура корпуса: W=55 оС, Z=85 оС.

Хотя эти процессоры по интерфейсу идентичны процессорам Enhanced Am486, их удастся использовать далеко не на всех системных платежах 486. Иногда причина кроется в версии BIOS, замена которой приводит к желаемому результату.

Иногда приходится снижать коэффициент умножения (если на плате есть джампер, позволяющий подать низкий уровень на вывод CLKMUL). Правда, при этом процессор становится аналогом DX-100 или DX4-120 в зависимости от выбранной входной частоты.

Кроме процессоров Intel и AMD, с шиной 486 процессора имеются продукты и других фирм. К ним относятся следующие:

Процессоры фирмы Cyrix:

- Cx486DX имеет по сравнению с другими более эффективный FPU. Процессоры Cx486DX2-66 и Cx486DX4-100 имеют кэш с обратной записью (WB), по параметрам близки к соответствующим моделям AMD.

- CYRIX 5x86-100 и 5x86-120 по внутренней архитектуре приближаются к пятому поколению (имеют, например, динамическое предсказание ветвлений), но внешнюю шину 486 процессора с расширенным режимом (кэш работает с обратной записью). Их производительность существенно выше 486-х

процессоров Intel и AMD с такими же тактовыми частотами. Проблемы с установкой этого процессора обычно связана с отсутствием его поддержкой конкретной версией BIOS. Кроме того, с этим процессором могут «зависать» некоторые программы, в частности написанные с помощью системы Clipper. Фирма Сугіх объясняет это явление тем, что задержки, реализованные на программных циклах, в этом процессоре будут иметь существенно меньше значение, чем в процессорах четвертого поколения (обратная сторона предсказаний ветвлений).

Для «лечения» этого «недуга» предлагаются специальные программы-замедлители, очевидно, отключающие архитектурные «излишества», а, к примеру, для использования пакета 3D-Studio с данным процессором предлагаются Patch-файлы («заплатки»).

Процессоры фирмы IBM.

-486BL2, 486B13 (Blue Lighting - молния) – вариант 486SX с 2-3-кратным умножением частоты без Burst Mode, питание 3,3 В и пониженное потребление. За звучным названием не стоят какие-либо серьезные преимущества.

-Несмотря на обозначение, процессоры 486SLC и 486DLC предназначены для замены 386SX и 386DX соответственно – их корпус и интерфейс к стандартной шине 486 процессоров отношения не имеют.

-Процессоры фирмы Texas Instruments.

-TIDX2-80 и TIDX4-100 близки к аналогичным 486-м процессорам AMD.

Контрольные вопросы:

1. Назовите основные этапы развития современных микропроцессоров.
2. Какие существуют подходы к классификации микропроцессоров по типу архитектуры?
3. Какие модели процессоров наиболее популярны в настоящее время?
4. Чем они отличаются друг от друга?

Тема 13. Структура микропроцессора

1. Структура микропроцессора.
2. Интерфейсные шины и организация памяти.
3. Процессы выполнения команд и обработки данных.
4. Способы повышения производительности процессора.
5. Многопроцессорные системы. Параметры и характеристики процессоров современных ПК.
6. Обзор периферийных устройств ПК.

1 Структура микропроцессора

Логическая структура микропроцессора, т. е. конфигурация составляющих микропроцессор логических схем и связей между ними, определяется функциональным назначением. Именно структура задает состав логических блоков микропроцессора и то, как эти блоки должны быть связаны между собой, чтобы полностью отвечать архитектурным требованиям. Срабатывание электронных блоков микропроцессора в определенной последовательности приводит к выполнению заданных архитектурой микропроцессора функций, т. е. к реализации вычислительных алгоритмов. Одни и те же функции можно выполнить в микропроцессорах со структурой, отличающейся набором, количеством и порядком срабатывания логических блоков. Различные структуры микропроцессоров, как правило, обеспечивают их различные возможности, в том числе и различную скорость обработки данных. Логические блоки микропроцессора с развитой архитектурой показаны на рис. 1.

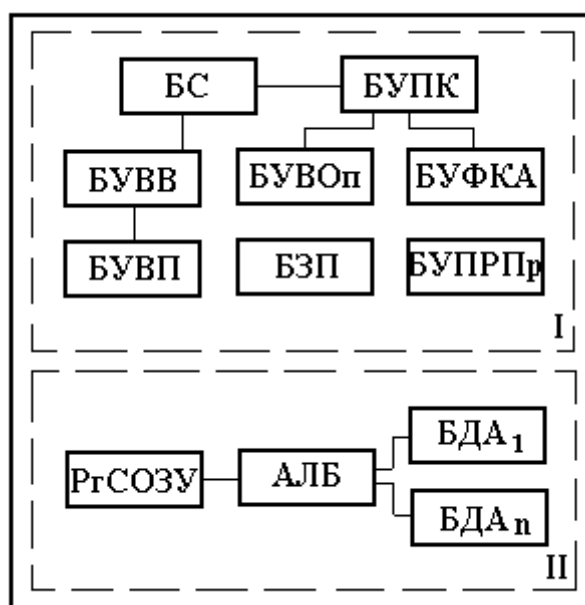


Рисунок 1 - Общая логическая структура микропроцессора: I - управляющая часть, II - операционная часть; БУПК - блок управления последовательностью команд; БУВОп - блок управления выполнением операций; БУФКА - блок управления формированием кодов адресов; БУВП -

блок управления виртуальной памятью; БЗП - блок защиты памяти; БУПРПр - блок управления прерыванием работы процессора; БУВВ - блок управления вводом/выводом; РгСОЗУ - регистровое сверхоперативное запоминающее устройство; АЛБ - арифметико-логический блок; БДА - блок дополнительной арифметики; БС - блок синхронизации.

При проектировании логической структуры микропроцессоров необходимо рассмотреть:

- 1) номенклатуру электронных блоков, необходимую и достаточную для реализации архитектурных требований;
- 2) способы и средства реализации связей между электронными блоками;
- 3) методы отбора если не оптимальных, то наиболее рациональных вариантов логических структур из возможного числа структур с отличающимся составом блоков и конфигурацией связей между ними.

При проектировании микропроцессора приводятся в соответствие внутренняя сложность кристалла и количество выводов корпуса. Относительный рост числа элементов по мере развития микроэлектронной технологии во много раз превышает относительное увеличение числа выводов корпуса, поэтому проектирование БИС в виде конечного автомата, а не в виде набора схем, реализующих некоторый набор логических переключательных функций и схем памяти, дает возможность получить функционально законченные блоки и устройства ЭВМ.

Использование микропроцессорных комплектов БИС позволяет создать микроЭВМ для широких областей применения вследствие программной адаптации микропроцессора к конкретной области применения: изменяя программу работы микропроцессора, изменяют функции информационно-управляющей системы. Поэтому за счет составления программы работы микропроцессоров в конкретных условиях работы определенной системы можно получить оптимальные характеристики последней.

Если уровень только программной "настройки" микропроцессоров не позволит получить эффективную систему, доступен следующий уровень проектирования - микропрограммный. За счет изменения содержимого ПЗУ или программируемой логической матрицы (ПЛМ) можно "настроиться" на более специфичные черты системы обработки информации. В этом случае частично за счет изменения микропрограмм затрагивается аппаратный уровень системы. Техничко-экономические последствия здесь связаны лишь с ограниченным вмешательством в технологию изготовления управляющих блоков микроЭВМ.

Изменение аппаратного уровня информационно-управляющей микропроцессорной системы, включающего в себя функциональные БИС комплекта, одновременно с конкретизацией микропрограммного и программного уровней позволяет наилучшим образом удовлетворить требованиям, предъявляемым к системе.

Решение задач управления в конкретной системе чисто аппаратными средствами (аппаратная логика) дает выигрыш в быстродействии, однако

приводит к сложностям при модификации системы. Микропроцессорное решение (программная логика) является более медленным, но более гибким решением, позволяющим развивать и модифицировать систему. Изменение технических требований к информационно-управляющей микропроцессорной системе ведет лишь к необходимости перепрограммирования работы микропроцессора. Именно это качество обеспечивает высокую логическую гибкость микропроцессоров, определяет возможность их широкого использования, а значит и крупносерийного производства.

2 Интерфейсные шины и организация памяти

Другое направление совершенствования *архитектуры* персонального компьютера связано с максимальным ускорением обмена информацией с системной памятью. Именно из системной памяти компьютер читает все исполняемые команды, и в системной же памяти он хранит данные. То есть больше всего обращений процессор совершает именно к памяти. Ускорение обмена с памятью приводит к существенному ускорению работы всей системы в целом. Но при использовании для обмена с памятью системной магистрали приходится учитывать скоростные ограничения магистрали. Системная магистраль должна обеспечивать сопряжение с большим числом устройств, поэтому она должна иметь довольно большую протяженность; она требует применения входных и выходных буферов для согласования с линиями магистрали. Циклы обмена по системной магистрали сложны, и ускорять их нельзя. В результате существенного ускорения обмена процессора с памятью по магистрали добиться невозможно.

Разработчиками был предложен следующий подход. Системная память подключается не к системной магистрали, а к специальной высокоскоростной шине, находящейся «ближе» к процессору, не требующей сложных буферов и больших расстояний. В таком случае обмен с памятью идет с максимально возможной для данного процессора скоростью, и системная магистраль не замедляет его. Особенно актуальным это становится с ростом быстродействия процессора (сейчас тактовые частоты процессоров персональных компьютеров достигают 1—3 ГГц).

Таким образом, структура персонального компьютера из одношинной, применявшейся только в первых компьютерах, становится трехшинной (рисунок 2).

Назначение шин следующее:

- к локальной шине подключаются центральный процессор и *кэш*-память (быстрая буферная память);
- к шине памяти подключается оперативная и постоянная память компьютера, а также контроллер системной шины;
- к системной шине (магистрали) подключаются все остальные устройства компьютера.

Все три шины имеют адресные линии, линии данных и управляющие сигналы. Но состав и назначение линий этих шин не совпадают между собой,

хотя они и выполняют одинаковые функции. С точки зрения процессора, системная шина (магистраль) в системе всего одна, по ней он получает данные и команды и передает данные как в память, так и в устройства ввода/вывода. Временные задержки между системной памятью и процессором в данном случае минимальны, так как локальная шина и шина памяти соединены только простейшими быстродействующими буферами. Еще меньше задержки между процессором и кэш-памятью, подключаемой непосредственно к локальной шине процессора и служащей для ускорения обмена процессора с системной памятью. Как правило, в компьютере применяются две и более системные шины, например, EISA, PCI и PCI Express. Каждая из них имеет свой собственный контроллер, и работают они параллельно. В этом случае получается многшинная структура компьютера, но на этом мы остановимся в лекции посвященной интерфейсам компьютера.

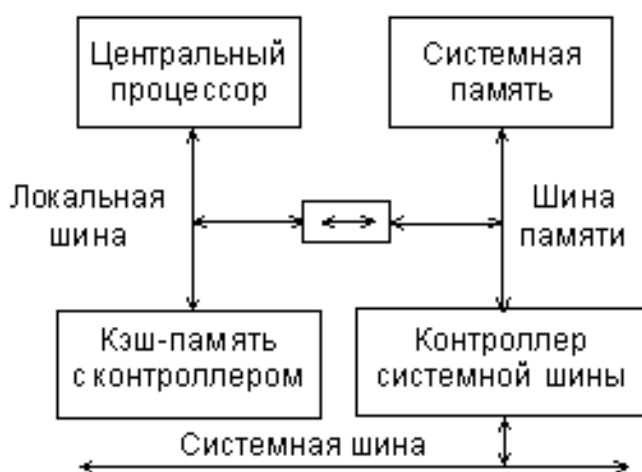


Рисунок 2 - Организация связей в случае трехшинной структуры

Адресное пространство МП состоит из множества ячеек памяти ОЗУ, из которых он может брать информацию или засылать ее. Как говорилось выше, начиная с 4-го поколения доминирует байтовая организация памяти, и минимально адресуемой единицей является байт. Например, для ОЗУ емкостью 1024 Кбайта адреса байтов таковы:

00000 00001 ... FFFFF

(как принято, записываем их в шестнадцатеричной системе; адрес последнего байта есть $1024 \cdot 1024 - 1 = 1\ 048\ 576 - 1 = \text{FFFFF}$). Длина же ячейки («машинного слова») может быть как один, так и несколько байтов в зависимости от типа процессора и команды, обрабатывающей соответствующую информацию.

При обмене информацией с памятью процессор обращается к ячейкам ОЗУ по их номерам (адресам). Способы задания требуемых адресов в командах ЭВМ принято называть *методами адресации*. От видов и разнообразия методов адресации существенно зависит эффективность работы программы с данными, особенно если последние организованы в определенную структуру.

Для того, чтобы процессор мог извлечь данные из ячейки ОЗУ или поместить их туда, необходимо где-то задать требуемый адрес. Если адрес находится в самой команде, то мы имеем дело с *прямой адресацией*. Поскольку при подобном способе слишком сильно возрастает длина команды, то, чтобы избежать этого неприятного эффекта, при обращении к ОЗУ процессор использует метод *косвенной адресации*. Идея состоит в том, что адрес памяти предварительно заносится в один из регистров МП, а в команде содержится лишь ссылка на этот регистр. Если учесть, что при хранении адреса в регистре его еще очень удобно модифицировать (скажем, циклически увеличивая на заданную величину), становится понятным, почему косвенная адресация нашла такое широкое применение.

Приведем описание наиболее распространенных вариантов ссылок на исходную информацию (учитывая, что терминология для разных МП может различаться, названия методов адресации не приводятся):

- 1) данные находятся в одном из регистров МП;
- 2) данные входят непосредственно в состав команды, т.е. размещаются после кода операции (операции с константой);
- 3) данные находятся в ячейке ОЗУ, адрес которой содержится в одном из регистров МП;
- 4) данные находятся в ячейке ОЗУ, адрес которой вычисляется по формуле *адрес = базовый адрес + смещение*.

Базовый адрес хранится в одном из регистров МП и является начальной точкой массива данных. Смещение может быть как некоторой константой, так и содержимым другого регистра. Часто такой способ доступа к ОЗУ называют *индексным*, так как это похоже на нахождение элемента в одномерном массиве по его индексу.

Следует подчеркнуть, что здесь описаны лишь наиболее общие методы адресации. У конкретных моделей МП существуют некоторые особенности адресации ОЗУ. Кроме того, имеющиеся методы адресации могут быть комбинированными. Так, например, в процессорах семейства PDP возможна двойная косвенная адресация: данные хранятся в ячейке ОЗУ, адрес которой хранится в ячейке, адрес которой находится в указанном регистре.

Методы адресации могут быть и более экзотическими. Рассмотрим, например, широко распространенный *сегментный* способ, принятый в процессорах фирмы «Intel». Известный американский программист Питер Нортон метко назвал этот способ словом «клудж» (от английского «kludge» - приспособление для временного устранения проблемы). Сегментный метод адресации был предложен для первого 16-разрядного МП 8086 для того, чтобы, используя 16-разрядные регистры, можно было получить 20-разрядный адрес и тем самым расширить максимально возможный объем ОЗУ с 64 кбайт до 1 Мбайта. Суть метода состоит в том, что адрес ОЗУ вычисляется как сумма двух чисел (сегмента и смещения), причем одно из них сдвинуто влево на 4 двоичных разряда, т.е. умножено на 16. Пусть, например, сегмент в шестнадцатеричном виде равен A000, а смещение - 1000. Общепринятая запись такого адреса имеет вид A000: 1000. Итоговый адрес равен

$$\begin{array}{r} A0000 \\ + 1000 \\ \hline A1000 \end{array}$$

«Такой способ адресации более 64 кбайт памяти кажется довольно странным, однако он работает», - не без некоторой иронии замечает по этому поводу Питер Нортон.

Добавим, что, помимо всего прочего, при сегментной адресации один и тот же адрес ОЗУ может быть представлен несколькими комбинациями чисел (в самом деле, одну и ту же сумму можно получить разными способами)

Поскольку современные интелловские процессоры, начиная с 80386, стали 32-разрядными, их регистров хватает, чтобы адресовать до 4 Гбайт памяти. Сегментный способ при этом становится излишним, хотя и сохраняется как вариант ради обеспечения программной совместимости с предыдущими моделями.

Нам осталось рассмотреть еще один очень специфический, но интересный и часто используемый на практике способ адресации данных в ОЗУ. Речь пойдет о работе со стеком. Сам термин происходит от английского слова «stack», имеющего множество значений, причем ни одно из них не подходит достаточно точно. *Стек* - это неявный способ адресации, при котором информация записывается и считывается только последовательным образом с использованием, так называемого, указателя стека.

Обратимся к примеру. Пусть требуется на время сохранить значения трех целочисленных переменных N1, N2 и N3, а затем их все сразу восстановить. Попробуем воспользоваться для этого стековой памятью. Прежде всего, запомним, что стек всегда имеет единственный вход и выход информации – для хранения его адреса и нужен указатель стека. Пусть для определенности он сейчас содержит адрес 2006 (рис. 3, а). Тогда по команде «записать в стек N1» (обратите внимание, что в команде не фигурирует явно ни адрес ОЗУ, ни регистр МП!) процессор проделает следующее:

- 1) уменьшит указатель стека на 2 (целое число занимает в памяти 2 байта);
- 2) запишет N1 по полученному адресу 2004 (рис. 3, б).

Аналогично при выполнении команд «записать в стек N2» и «записать в стек N3» значения этих переменных попадут в ячейки 2002 и 2000, причем указатель стека станет равным 2000 (рис. 3, в).

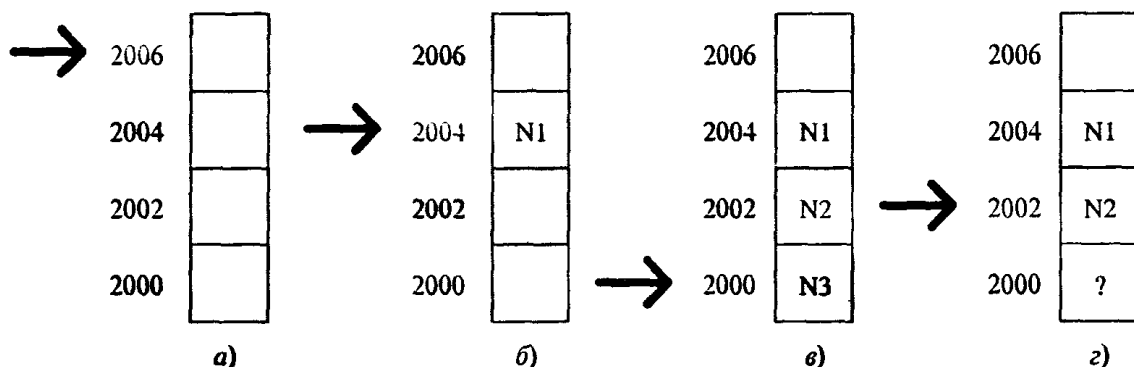


Рисунок 3 - Пример для изучения принципа работы стека

Теперь займемся извлечением информации. Как видно из рис. 3, в, указатель сейчас «направлен» на значение переменной N3, а значит считывать придется, начиная с него. Выполним команду «прочитать из стека N3». При этом процессор

- 1) считает в N3 значение из стека;
- 2) увеличит указатель на 2 (рис. 3, г).

Аналогично прочитаем N2 и N1, после чего стек опустеет и вернется к начальному состоянию, изображенному на рис. 3, а.

Примечание. Обратим внимание на то, что значение в стеке после считывания, конечно же, не исчезает, как это условно показано на рис. 3, г, но есть причины полагать, что мы его там можем больше не увидеть. Дело в том, что процессор имеет право иногда временно использовать стек для своих «внутренних» нужд. При этом некоторые ячейки, адреса которых меньше текущего указателя стека, естественно, изменятся. Следовательно, во избежание неприятностей лучше всегда считать, что после считывания информации в стековой памяти она пропадает.

Подводя итоги рассмотрения примера, обратим внимание на следующее. Та информация, которая заносится в стек первой, извлекается последней и наоборот. Обычно об организации стека говорят: «первым пришел - последним ушел». Удобно представить себе аналогию с детской пирамидкой, нижнее кольцо которой невозможно снять до тех пор, пока не будут сняты все остальные.

Еще раз подчеркнем, что в командах работы со стеком адрес ОЗУ не фигурирует в явном виде. Но при этом молчаливо предполагается, что указатель стека уже задан. Как правило, инициализацию указателя стека компьютер делает сам, но не мешает поинтересоваться, так ли это в вашем конкретном случае и достаточно ли памяти под стек имеется при этом. Если указатель стека определен неправильно, то запись в стек может разрушить полезную информацию в ОЗУ и вызвать непредсказуемые последствия.

3 Процессы выполнения команд и обработки данных

В микропроцессорах используют два метода выработки совокупности функциональных управляющих сигналов: программный и микропрограммный.

Выполнение операций в машине сводится к элементарным преобразованиям информации (передача информации между узлами в блоках, сдвиг информации в узлах, логические поразрядные операции, проверка условий и т.д.) в логических элементах, узлах и блоках под воздействием функциональных управляющих сигналов блоков (устройств) управления. Элементарные преобразования, неразложимые на более простые, выполняются в течение одного такта сигналов синхронизации и называются микрооперациями.

В аппаратных (схемных) устройствах управления каждой операции соответствует свой набор логических схем, вырабатывающих определенные

функциональные сигналы для выполнения микроопераций в определенные моменты времени. При этом способе построения устройства управления реализация микроопераций достигается за счет однажды соединенных между собой логических схем, поэтому ЭВМ с аппаратным устройством управления называют ЭВМ с жесткой логикой управления. Это понятие относится к фиксации системы команд в структуре связей ЭВМ и означает практическую невозможность каких-либо изменений в системе команд ЭВМ после ее изготовления.

При микропрограммной реализации устройства управления в состав последнего вводится ЗУ, каждый разряд выходного кода которого определяет появление определенного функционального сигнала управления. Поэтому каждой микрооперации ставится в соответствие свой информационный код - микрокоманда. Набор микрокоманд и последовательность их реализации обеспечивают выполнение любой сложной операции. Набор микроопераций называют микропрограммами. Способ управления операциями путем последовательного считывания и интерпретации микрокоманд из ЗУ (наиболее часто в виде микропрограммного ЗУ используют быстродействующие программируемые логические матрицы), а также использования кодов микрокоманд для генерации функциональных управляющих сигналов называют микропрограммным, а микроЭВМ с таким способом управления - микропрограммными или с хранимой (гибкой) логикой управления.

К микропрограммам предъявляют требования функциональной полноты и минимальности. Первое требование необходимо для обеспечения возможности разработки микропрограмм любых машинных операций, а второе связано с желанием уменьшить объем используемого оборудования. Учет фактора быстродействия ведет к расширению микропрограмм, поскольку усложнение последних позволяет сократить время выполнения команд программы.

Преобразование информации выполняется в универсальном арифметико-логическом блоке микропроцессора. Он обычно строится на основе комбинационных логических схем.

Для ускорения выполнения определенных операций вводятся дополнительно специальные операционные узлы (например, циклические сдвигатели). Кроме того, в состав микропроцессорного комплекта (МПК) БИС вводятся специализированные оперативные блоки арифметических расширителей.

Операционные возможности микропроцессора можно расширить за счет увеличения числа регистров. Если в регистровом буфере закрепление функций регистров отсутствует, то их можно использовать как для хранения данных, так и для хранения адресов. Подобные регистры микропроцессора называются регистрами общего назначения (РОН). По мере развития технологии реально осуществлено изготовление в микропроцессоре 16, 32 и более регистров.

В целом же, принцип микропрограммного управления (ПМУ) включает следующие позиции:

1) любая операция, реализуемая устройством, является последовательностью элементарных действий - микроопераций;

2) для управления порядком следования микроопераций используются логические условия;

3) процесс выполнения операций в устройстве описывается в форме алгоритма, представляемого в терминах микроопераций и логических условий, называемого микропрограммой;

4) микропрограмма используется как форма представления функции устройства, на основе которой определяются структура и порядок функционирования устройства во времени.

ПМУ обеспечивает гибкость микропроцессорной системы и позволяет осуществлять проблемную ориентацию микро- и миниЭВМ.

4 Способы повышения производительности процессора

Обработка команды, или цикл процессора, может быть разделена на несколько основных этапов, которые можно назвать микрокомандами. Каждая операция требует для своего выполнения времени, равного такту генератора процессора. Конвейеризация осуществляет многопоточную параллельную обработку команд, так что в каждый момент одна из команд считывается, другая декодируется и т. д., и всего в обработке одновременно находится пять команд. Таким образом, на выходе конвейера на каждом такте процессора появляется результат обработки одной команды (одна команда в один такт). Первая инструкция может считаться выполненной, когда завершат работу все пять микрокоманд. Такая технология обработки команд носит название конвейерной (pipeline) обработки. Каждая часть устройства называется ступенью конвейера, а общее число ступеней — длиной линии конвейера. С ростом числа линий конвейера и увеличением числа ступеней на линии увеличивается пропускная способность процессора при неизменной тактовой частоте.

Процессоры с несколькими линиями конвейера получили название суперскалярных. Pentium — первый суперскалярный процессор Intel. Здесь две линии, что позволяет ему при одинаковых частотах быть вдвое производительней i80486, выполняя сразу две команды за такт. Во многих вычислительных системах наряду с конвейером команд используются конвейеры данных. Сочетание этих двух конвейеров дает возможность достичь очень высокой производительности на определенных классах задач, особенно если используется несколько различных конвейерных процессоров, способных работать одновременно и независимо друг от друга.

Матричные и векторные процессоры. В отличие от скалярных и даже суперскалярных процессоров данные устройства манипулируют массивами данных и предназначены для обработки изображений, матриц и массивов данных. Частным случаем векторного процессора является процессор изображений, который служит для обработки сигналов, поступающих от датчиков—формирователей изображения.

Матричный процессор имеет архитектуру, рассчитанную на обработку числовых массивов, например матриц. Архитектура процессора включает в себя матрицу процессорных элементов, например 64x64, работающих одновременно. Постпроцессор предназначен для реализации некоторых специальных функций, например управления базой данных.

Векторный процессор обеспечивает параллельное выполнение операции над массивами данных, векторами. Он характеризуется специальной архитектурой, построенной на группе параллельно работающих процессорных элементов. Максимальная скорость передачи данных в векторном формате может составлять 64 Гбайт/с, что на два порядка быстрее, чем в скалярных машинах.

SIMD – команды. Г. Флинном в 1966 г. была предложена классификация ЭВМ и вычислительных систем (в основном суперкомпьютеров), основанная на совместном рассмотрении потоков команд и данных. В процессорах таких известных производителей, как Intel и AMD, все более полно используются некоторые из этих архитектурных наработок.

В общем случае архитектура SIMD (ОКМД) предполагает создание структур векторной или матричной обработки. Системы этого типа обычно строятся как однородные, т. е. процессорные элементы, входящие в систему, идентичны, и все они управляются одной и той же последовательностью команд. Однако каждый процессор обрабатывает свой поток данных. Под эту схему хорошо подходят задачи обработки матриц или векторов (массивов), задачи решения систем линейных и нелинейных, алгебраических и дифференциальных уравнений, задачи теории поля и др.

В процессоре Pentium MMX были применены элементы SIMD-команд для обработки мультимедийных данных (видеокодирование, шкалирование, экстраполяция). При этом достигалось общее повышение производительности на 10—20 %, а в программах обработки мультимедиа - до 60 %.

В процессорах Pentium II, III введено 50—70 новых команд, названных Streaming SIMD Extensions (SSE). Процессор содержит "битовые регистры", позволяющие осуществлять за один такт до четырех операций с плавающей точкой.

В дальнейшем было введено еще 76 SIMD-команд и модернизированы 68 имеющихся команд, что получило название SSE2 (Pentium IV Northwood). Здесь 128-битовые регистры обеспечивали обработку как чисел. Таким образом, SSE2 более гибок, позволяя добиваться роста в производительности. Однако использование нового набора команд требует специальной оптимизации программ. AMD также реализует SSE2 в своем новом семействе процессоров Hammer. Процессор Pentium IV Prescott содержит еще на 13 SSE-команд больше (SSE3).

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

1. множественное предсказание ветвлений;
2. анализ потока данных;
3. спекулятивное (по предположению) исполнение.

Множественное предсказание ветвлений. Предсказывает прохождение программы по нескольким ветвям. Процессор может предвидеть разделение потока команд, используя алгоритм множественного предсказания ветвлений. С большой точностью (более 90 %) он предсказывает, в какой области памяти можно найти следующие команды. Это оказывается возможным, поскольку в процессе исполнения команды процессор просматривает программу на несколько шагов вперед. Этот метод позволяет увеличить загруженность процессора.

Анализ потока данных. Анализирует и составляет график исполнения команд в оптимальной последовательности, независимо от порядка их следования в тексте программы. Используя анализ потока данных, процессор просматривает декодированные команды и определяет, готовы ли они к непосредственному исполнению или зависят от результата других команд. Далее процессор определяет оптимальную последовательность выполнения и исполняет команды наиболее эффективным образом.

Спекулятивное выполнение. Повышает скорость выполнения, просматривая программу вперед и исполняя те команды, которые необходимы. Процессор выполняет команды (до пяти команд одновременно) по мере их поступления в оптимизированной последовательности (спекулятивно). Поскольку выполнение команд происходит на основе предсказания ветвлений, результаты сохраняются как «спекулятивные». На конечном этапе порядок команд восстанавливается.

Технология Hyper-Threading (HT). Эта архитектура реализует разделение времени на аппаратном уровне, разбивая физический процессор на два логических процессора, каждый из которых использует ресурсы чипа — ядро, кэш-память, шины, исполнительное устройство. Ядро процессора выполняет два процесса одновременно.

Специалисты Intel оценивают повышение эффективности в 30% при использовании на HT-процессорах многопрограммных ОС и обычных прикладных программ.

Технология 3DNow! Улучшая возможности процессора обращаться с вычислениями с плавающей запятой, 3DNow! технология устранила растущую разницу в производительности процессора и графического акселератора и устранила узкое место в начале графического конвейера. Это очистило путь для быстрорастущей производительности 3D и мультимедиа.

Эти возможности впервые были реализованы в процессорах AMD K6-2 (май 1998 г.), и Intel Pentium III Katmai (первая половина 1999 г.). Технологии 3DNow! используются в широком диапазоне приложений — игры, Web-сайты VRML, автоматизированное проектирование, распознавание речи и программное обеспечение декодирования DVD. Производительность в дальнейшем была повышена за счет использования DirectX 6.0 Microsoft, представленного летом 1998 г. Последующие версии OpenGL API были также

оптимизированы для 3DNow!. К концу марта 1999 г. ПЭВМ, основанные на технологии 3DNow!, достигли приблизительно 14 млн систем во всем мире.

5 Многопроцессорные системы. Параметры и характеристики процессоров современных ПК

Многопроцессорность (Мультипроцессорность, Многопроцессорная обработка, англ. *Multiprocessing*) — использование пары или большего количества физических процессоров в одной компьютерной системе. Термин также относится к способности системы поддерживать больше чем один процессор и/или способность распределить задачи между ними. Существует много вариантов данного понятия, и определение многопроцессорности может меняться в зависимости от контекста, главным образом в зависимости от того, как определены процессоры (много ядер в одном кристалле, множество чипов в одном корпусе, множество корпусов в одном системном модуле, и т. д.).

Многопроцессорностью иногда называют выполнение множественных параллельных программных процессов в системе в противоположность выполнению одного процесса в любой момент времени. Однако термины многозадачность или мультипрограммирование являются более подходящими для описания этого понятия, которое осуществлено главным образом в программном обеспечении, тогда как многопроцессорная обработка является более соответствующей, чтобы описать использование множественных аппаратных процессоров. Система не может быть и многопроцессорной и мультипрограммированной, только одной из двух, или ни той и ни другой.

Процессорная симметричность.

В многопроцессорной системе все центральные процессоры могут быть равными, или некоторые могут быть зарезервированы для особых целей. Комбинация конструктивных соображений программного обеспечения аппаратной и операционной системы определяет симметрию (или отсутствие её) в данной системе. Например, аппаратные или программные соображения могут потребовать, чтобы только один центральный процессор отвечал на все аппаратные прерывания, тогда как вся другая работа в системе может быть распределена одинаково среди процессоров; или выполнение кода привилегированного режима может быть ограничено только одним процессором (или определенным процессором, или только один процессор за один раз), тогда как код непривилегированного режима может быть выполнен на любой комбинации процессоров. Часто многопроцессорные системы проще проектировать, если введены такие ограничения, но они имеют тенденцию быть менее эффективными, чем системы, в которых используются все центральные процессоры.

Системы, которые обрабатывают все центральные процессоры одинаково, называют системами с симметричной многопроцессорной обработкой (SMP). В системах, где все центральные процессоры не равны, системные ресурсы могут быть разделены многими способами, в том числе асимметричной многопроцессорной обработкой (ASMP), многопроцессорной обработкой с

неоднородным доступом к памяти (NUMA) и кластеризованной многопроцессорной обработкой (q.v.).

Потоки команд и данных.

В многопроцессорной обработке процессоры могут использоваться для выполнения одной последовательности команд во множественных контекстах (единственная машинная команда, множественные данные или SIMD, часто используемый в векторной обработке), множественные последовательности команд в единственном контексте («множественный поток команд, одиночный поток данных») или Архитектура MISD, используемая для избыточности в отказоустойчивых системах и иногда применяемая, чтобы описать конвейерные процессоры или гиперпоточность), или множественные последовательности команд во множественных контекстах («множественный поток команд, множественные поток данных») или MIMD).

Соединения процессоров.

Многопроцессорные системы с сильной связью (англ. *Tightly-coupled multiprocessor systems*) содержат несколько процессоров, которые подключены на шинном уровне. Эти процессоры могут иметь доступ к центральной разделяемой памяти (SMP или UMA), или могут участвовать в иерархии памяти и с локальной и с разделяемой памятью (NUMA). IBM p690 Regatta является примером мощной системы SMP. Процессоры Intel Xeon доминировали над многопроцессорным рынком для деловых PC и были единственной x86-опцией до выпуска линейки процессоров AMD Opteron в 2004 году. Обе линейки процессоров имели свой собственный встроенный кэш, но по разному обеспечивают доступ к разделяемой памяти: процессоры Xeon через общий канал, а процессоры Opteron через независимые магистрали к системной оперативной памяти.

Многопроцессорные (многоядерные) чипы, включает больше одного процессора, помещённого в однокристалльную схему, и могут считаться самой предельной формой многопроцессорной обработки с сильной связью. Мэйнфреймовые системы со множеством процессоров — часто являются системами с сильной связью.

Многопроцессорные системы с гибкой связью (англ. *Loosely-coupled multiprocessor systems*), часто называемые кластерами, основаны на множественных автономных одиночных или двойных компьютерах, связанных через высокоскоростную систему связи (например, Gigabit Ethernet). Кластер Беовульфа под управлением Linux — пример гибко связанной системы.

Системы с сильной связью работают лучше и физически они меньше, чем гибко связанные системы, но исторически потребовали больших начальных инвестиций и могут быстро амортизовываться; узлы в гибко связанной системе — обычно недорогие компьютеры и могут быть использованы как независимые машины после удаления из кластера.

Системы с сильной связью имеют тенденцию быть намного больше энергоэффективными, чем кластеры. Значительная энергоэффективность достигается за счёт того, что компоненты в таких системах заранее проектируются для работы в конкретной системе, в то время как гибко

связанные системы зачастую используют компоненты спроектированные для работы в более широком классе систем.

Программные реализации.

Многопроцессорная обработка с SISD.

В компьютере с одиночным потоком команд и одиночным потоком данных один процессор последовательно обрабатывает команды; каждая машинная команда обрабатывает один элемент данных. Пример — фон-неймановская архитектура.

Многопроцессорная обработка SIMD.

В компьютере с одиночным потоком команд и множественным потоком данных один процессор обрабатывает поток команд, каждая из которых может выполнить параллельные вычисления на множестве данных.

Многопроцессорная обработка SIMD хорошо подходит для параллельной или векторной обработки, в которой большой набор данных может быть разделён на части, которые обрабатываются идентичными, но независимым операциями. Одиночный поток команд направляет^[куда?] операцию модулей мультипрограммирования для выполнения однотипных манипуляций одновременно на потенциально большом количестве данных.

Для определённых типов вычислительных приложений этот тип архитектуры может дать значительное увеличение производительности с точки зрения затраченного времени. Однако, недостаток этой архитектуры состоит в том, что большая часть системы начинает простаивать при выполнении программ или системных задач, которые не могут быть разделены на модули (подзадачи), которые могут быть обработаны параллельно.

Кроме того, программы должны быть тщательно и специально написаны, чтобы иметь возможность максимально задействовать возможности архитектуры. Часто применяются специальные оптимизирующие компиляторы, спроектированные чтобы создать код специально для использования в этой среде. Некоторые компиляторы в этой категории обеспечивают специальные конструкции или расширения, чтобы позволить программистам непосредственно определять операции, которые будут выполнены параллельно (например, операторы DO FOR ALL в Фортране, используемого на ILLIAC IV, который был многопроцессорным суперкомпьютером с SIMD-архитектурой).

Многопроцессорная обработка SIMD находит широкое применение в некоторых областях, таких как компьютерное моделирование, но малополезна в универсальном настольном компьютере и бизнес-задачах.

Многопроцессорная обработка MISD.

Многопроцессорная обработка с множественным потоком команд и одиночным потоком данных предлагает главным образом преимущество избыточности, так как модули мультипрограммирования выполняют одни задачи на одних данных, уменьшая возможности неправильных результатов, если один из модулей выходит из строя. Архитектура MISD позволяет сравнивать результаты вычислений в целях обнаружения отказов. Кроме избыточности и отказоустойчивости у этого типа многопроцессорной

обработки немного преимуществ. К тому же он весьма дорог. Он не увеличивает производительность.

Многопроцессорная обработка MIMD.

Архитектура многопроцессорной обработки MIMD является подходящей для большого числа разнообразных задач, в которых реализовано полностью независимое и параллельное выполнение команд, касающихся различных наборов данных. По этой причине и потому что это просто осуществить, MIMD преобладает в многопроцессорной обработке.

Обработка разделена на несколько потоков, каждый с собственным аппаратным состоянием процессора, в рамках единственного определённого программным обеспечением процесса или в пределах множественных процессов. Поскольку система имеет несколько потоков, ожидающих выполнения (системные или пользовательские потоки), эта архитектура эффективно использует аппаратные ресурсы.

В MIMD могут возникнуть проблемы взаимной блокировки и состязания за обладание ресурсами, так как потоки, пытаясь получить доступ к ресурсам, могут столкнуться непредсказуемым способом. MIMD требует специального кодирования в операционной системе компьютера, но не требует изменений в прикладных программах, кроме случаев когда программы сами используют множественные потоки (MIMD прозрачен для однопоточных программ под управлением большинства операционных систем, если программы сами не отказываются от управления со стороны ОС). И системное и пользовательское программное обеспечение, возможно, должны использовать программные конструкции, такие как семафоры, чтобы препятствовать тому, чтобы один поток вмешался в другой, в случае если они содержат ссылку на одни и те же данные. Такое действие увеличивает сложность кода, снижает производительность и значительно увеличивают количество необходимого тестирования, хотя обычно не настолько чтобы свести на нет преимущества многопроцессорной обработки.

Подобные конфликты могут возникнуть на аппаратном уровне между процессорами, и должен обычно решаться аппаратными средствами, или с комбинацией программного обеспечения и оборудования.

6 Обзор периферийных устройств ПК

Современные персональные компьютеры обычно имеют в своем распоряжении множество периферийных устройств.

Периферийные устройства – это любые дополнительные и вспомогательные устройства, которые подключаются к ПК для расширения его функциональных возможностей.

Рассмотрим некоторые из периферийных устройств.

Принтер (print - печатать) – устройство для вывода на печать текстовой и графической информации. Принтеры, как правило, работают с бумагой формата А4 или А3. Наиболее распространены на сегодняшний день лазерные и струйные принтеры, матричные принтеры уже вышли из обихода.

В матричных принтерах печатающая головка состояла из ряда тонких металлических иголок, которые при движении вдоль строки в нужный момент ударяли через красящую ленту, и тем самым обеспечивали формирование символов и изображения. Матричные принтеры обладали низкими скоростью и качеством печати.

В струйных принтерах краска под давлением выбрасывается из отверстий (сопел) в печатающей головке и затем прилипает к бумаге. При этом формирование изображения происходит как бы из отдельных точек - "клякс". Для струйных принтеров характерна высокая стоимость расходных материалов.

В лазерных принтерах луч лазера, пробегаая по барабану, электризует его, а наэлектризованный барабан притягивает частицы сухой краски, после чего изображение переносится с барабана на бумагу. Далее лист бумаги проходит через тепловой барабан и под действием тепла краска фиксируется на бумаге. Лазерные принтеры обладают высокими скоростью и качеством печати.

Плоттер (графопостроитель) – устройство для вывода на бумагу больших рисунков, чертежей и другой графической информации. Плоттер может выводить графическую информацию на бумагу формата А2 и больше. Конструктивно в нем может использоваться или барабан рулонной бумаги, или горизонтальный планшет.

Сканер (scanner) – устройство, позволяющее вводить в компьютер графическую информацию. Сканер при движении по картинке (лист текста, фотография, рисунок) преобразует изображение в числовой формат и отображает его на экране. Затем эту информацию можно обработать с помощью компьютера.

Манипулятор мышь (mouse) – устройство, облегчающее ввод информации в компьютер.

Дисковод CD-ROM – устройство для чтения информации, записанной на лазерных компакт-дисках (CD ROM – Compact Disk Read Only Memory, что в переводе означает компакт-диск с памятью только для чтения). На компакт-дисках можно хранить большое количество информации (до 650 Мбайт). Такие диски используются для хранения справочной информации, больших энциклопедий, баз данных, музыки, видеоинформации и т.д.

Основной показатель для дисковода CD-ROM – это скорость считывания информации с компакт-диска.

Дисковод DVD является дальнейшим развитием лазерных технологий. В нем применяется усовершенствованная технология использования лазерного луча для записи и чтения информации с компакт-дисков. Аббревиатура DVD означает Digital Video Disk (цифровой видеодиск) или в другой трактовке - Digital Versatile Disk (цифровой многоцелевой диск).

В отличие от дисков CD-ROM диски DVD могут использовать для работы обе поверхности. Причем технология позволяет записывать на каждой из сторон два слоя данных.

Контрольные вопросы:

1. Поясните структуру микропроцессора.
2. Какие факторы рассматриваются при проектировании логической структуры микропроцессоров?
3. Сколько шин применяются в современных МП?
4. Их назначение.
5. Какие существуют методы адресации данных?
6. Какие методы используются для выработки совокупности функциональных управляющих сигналов?
7. Как можно расширить операционные возможности микропроцессора?
8. В чем разница между матричными и векторными процессорами?
9. Какие используются сегодня способы повышения производительности процессора?

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Сергеев Н.П., Вашкевич Н.П. Основы вычислительной техники. М.: «Мир», 2012.
2. Гук М. Процессоры фирмы Intel от 8086 до PENTIUM V. СПб.: Питер-Пресс, 2008.
3. Гук М. Аппаратные средства IBM PC. СПб.: Питер, 2000.
4. Пухальский Г.И., Новосельцева Т.Я. Цифровые устройства: Учебное пособие для втузов. СПб.: Политехника, 2006.
5. Самофалов К.Г., Корнейчук И.В., Тарасенко В.П. Цифровые электронные вычислительные машины. Киев: Вища школа, 2015.
6. Гук М., Юров В. Pentium 4 Athlon и Duron. СПб.: Питер, 2011.
7. Томпсон Р.Б., Томпсон Б.Ф. Железо ПК: Энциклопедия. СПб.: Питер, 2003.
8. Дейтел Х., Дейтел П. Как программировать на Си. М.:Бином, 2000.
9. Соломонов Л.А. Персональные автоматизированные информационные системы. М.: Высшая школа, 1998.
10. Гасов В.М. Отображение информации. М.: Высшая школа, 2003.
11. Сальников Ю.В. Средства общения с ЭВМ. М.: Высшая школа, 2009.
12. Айден К. Аппаратные средства PC. Санкт-Петербург: БХВ, 2008.
13. Смоляров А.М. Системы отображения информации и инженерная психология. М.: Высшая школа, 2000.
14. Гук М., Юров В. Процессоры Pentium IV, Athlon и Duron. — СПб.: «Питер», 2001.
15. Есмагамбетов Б.С. Микропроцессорная техника. Учебное пособие. - КазАТК им. М.Тынышпаева, Шымкентский филиал, 2006.
16. Григорьев В. Л. Микропроцессор i 486: архитектура и программирование, в 4-х книгах. - М.: ТОО «Гранал», 1993.
17. Напрасник М.В. Микропроцессоры и микроЭВМ. – Москва: «Высшая школа», 1989 г.
18. Злобин В. К., Григорьев В.Л. Программирование арифметических операций в микропроцессоре. - М.: «Высшая школа», 1991.
19. Таненбаум Э. Архитектура компьютера. — СПб.: Питер, 2002.
20. Михаил Гук. От Intel 8086 до Intel 80586. – СП.: «Питер», 1991.
21. Токхайм Р. Л. Микропроцессоры: курс и упражнения. - М.: «Энергоатомиздат», 1987.
22. Нортон П. Персональный компьютер изнутри. - М.: «Бином», 2015.
23. Михаил Гук. Энциклопедия. Аппаратные средства IBM PC. - СП.: «Питер», 1999 г.
24. Бродин Б. Б., Шагурин И. И. – Микропроцессор Intel. – М.: «Диалог-МИФИ», 1993 г.
25. Юэн Ч. и др. Микропроцессорные системы и их применение при обработке сигналов. – Москва: «Радио и связь», 1986 г.

26. Стрыгин В.В., Щарев Л.С. Основы вычислительной микропроцессорной техники и программирования. - М.: «Высшая школа», 1989 г.
27. Микропроцессорные БИС и микроЭВМ / Васенков А.А., Воробьев Н.М. и др. - М.: «Советское Радио», 1980 г.
28. Горбунов Л.П., Панфилов Д.И., Преснухин Д.Л. Микропроцессоры. Основы построения микроЭВМ. - М.: «Высшая школа», 1984 г.
29. Григорьев В.Л. Программирование однокристалльных микропроцессоров. - М.: «Энергоатомиздат», 1987 г.
30. Мураховский В. Сборка. Настройка. Апгрейд ПК. - М.: «Десс», 1999 г.
31. Алексеенко А.Г. Проектирование радиоэлектронной аппаратуры на микропроцессорах: Программирование, типовые решения, методы отладки. - М.: «Радио и связь», 1984 г.
32. Юров В., Хорошенко С. Assembler. – СПб: «Питер», 1999 г.
33. Ермаков Н.Т. Информационные технологии. – Астана: «Фолиант», 2006 г.
34. Ямпольский В.С. Основы автоматики и электронно–вычислительной техники. Учебное пособие – М: «Просвещение», 1991 г.
35. Мняян М.Г. Физические принципы работы ЭВМ – М: «Просвещение», 1987 г.
36. Густоваров В.И. Ассемблер: программирование и анализ коррективности машинных программ – Киев: «Издательская группа ВНУ», 2000 г.
37. Электронный учебник: Ершова Н.Ю., Ивашенков О.Н., Курсков С.Ю. «Микропроцессоры».
38. Электронный ресурс: http://www.distedu.ru/mirror/_inform/conspect.narod.ru/doc_6.htm
39. Электронный ресурс: <http://zaz.gendocs.ru/docs/2800/index-570107.html?page=19>
40. Электронный ресурс: http://botanik.ucoz.com/index/perspektivy_razvitija_mikroprocessorov/0-129.
41. Электронный ресурс: http://life-prog.ru/view_articles.php?id
42. Электронный ресурс: <http://dfe.petrstu.ru/koi/posob/microcpu/proekt3.html>.

