

Введение

В последнее десятилетие развитие компьютерных технологий вступило в совершенно новую фазу. Вычислительные машины пришли во все сферы человеческой деятельности. Практически каждое предприятие, не зависимо от сферы деятельности, трудно представить без компьютерной техники.

Широкое распространение персональных компьютеров и микрокомпьютерных рабочих станций в бизнесе началось в начале 80-х годов. В настоящее время это оборудование представляет собой столь же неотъемлемую принадлежность любого офиса, как, например, телефон. До относительно недавних пор офисные локальные сети предоставляли в основном услуги соединения — они соединяли персональные компьютеры и терминалы с мэйнфреймами и системами промежуточного звена, на которых работали корпоративные приложения, а также обеспечивали соединение для рабочих групп на уровне отделов или цехов. В обоих случаях объемы передаваемых по сетям данных были невелики. Трафик в основном состоял из файлов и электронной почты. Локальные сети, предназначенные для обслуживания подобного трафика, а это в первую очередь Ethernet и Token Ring, отлично подходят для такого окружения.

В 90-е годы роль персонального компьютера (а вместе с ней и требования к локальной сети) изменилась благодаря двум важным тенденциям:

- Вычислительная мощь персональных компьютеров продолжает радовать экспоненциальным ростом. Все более мощные платформы поддерживают приложения, интенсивно использующие графику, и все усложняющиеся графические интерфейсы пользователя с операционными системами.

- Организации, занимающиеся административными информационными системами, увидели в локальных сетях жизнеспособные и важные вычислительные платформы. Начало этой тенденции положили системы «клиент-сервер», ставшие доминирующей архитектурой в деловой среде, а также более недавнее веяние — технология интранет.

Результатом этих тенденций стала необходимость обеспечения роста передаваемого по локальным сетям объема данных и, так как приложения стали более интерактивными, уменьшения времени задержки при передаче данных.

Так как при работе используется не одна, а несколько вычислительных машин (в некоторых случаях сотни и даже тысячи), существует острая необходимость объединения их в одно единое информационное пространство. Эту задачу с успехом решают сетевые технологии, позволяющие, в зависимости от сферы деятельности, географического положения и структуры предприятия, предложить десятки готовых решений.

При моделировании локальных вычислительных сетей многие специалисты не учитывают очень важные факторы, такие как: количество машин в сети, число пользователей, объемы передаваемой информации. Эти недочеты, допущенные на начальном этапе проектирования, приводят к неудовлетворительной работе сети, а иногда и полной неработоспособности и потери данных.

Даже правильно спроектированная сеть может постепенно выработать свой ресурс, ввиду своего постепенного расширения и увеличения нагрузки на оборудование.

В этих условиях крайне необходимо устройство, которое могло бы не только оценивать целостность сети и готовность оборудования для решения поставленных задач, но и могло оценить реальную загруженность как всей сети в целом, так и отдельных ее участков и даже конечного оборудования.

Учитывая эту необходимость, Я решил в моем дипломном проекте разработать такое устройство. При чем, принимая требования универсальности оборудования, мое устройство должно иметь стандартные интерфейсы, для установки в распространенное оборудование.

Глава 1

Обзор существующего положения микропроцессорной техники и сетевых технологий

Появление и развитие микропроцессоров ознаменовало собой начало новой эры не только в вычислительной технике, но и в области систем встроенного управления оборудованием — эры высокопроизводительных и надежных цифровых, микропроцессорных систем управления, интегрированных в рабочую машину, механизм, прибор, станок, робот, стиральную машину, принтер, плоттер, радиотелефон, автомобиль и т.д.

Компьютерная и связанная с ней микропроцессорная техника развиваются столь стремительно, что производительность процессоров удваивается каждые 8-10 месяцев при значительном росте степени интеграции, снижении габаритов, энергопотребления и цены.

Революция в управляющей электронике сопровождается революцией и в силовой электронике, что позволяет создавать интегрально-гибридные интеллектуальные электронные модули, а также конструктивно интегрировать в одном изделии рабочий орган механизма, силовой преобразователь, устройство управления, источник питания и датчики.

Встроенной системой управления будем называть систему управления, конструктивно интегрированную в оборудование. Обязательным компонентом такой системы является интерфейс с системой управления более высокой: уровня (промышленным программируемым контроллером или компьютером в промышленном исполнении), что позволяет решать задачи комплексной автоматизации с использованием заданного числа единиц технологического оборудования объединяя это оборудование в единую распределенную систему управления.

В зависимости от сложности решаемой задачи встроенная система управления может быть однопроцессорной или многопроцессорной (мультимикропроцессорной), одноуровневой или многоуровневой

Для многоуровневых систем управления в настоящее время четко прослеживается тенденция оформления систем управления среднего уровня в самостоятельном конструктиве для монтажа в стойки или в панели управления.

Возможен также вариант настольного или настенного монтажа. Такие устройства управления получили название промышленных рабочих станций, промышленных компьютеров, панельных компьютеров, промышленных программируемых контроллеров. При этом контроллеры нижнего уровня управления могут подключаться к системе среднего уровня управления несколькими способами: через стандартные последовательные или параллельные интерфейсы; путем установки на системную шину в качестве специализированных устройств сопряжения с объектом (модулей УСО).

Индустриальные компьютеры имеют, как правило, развитый набор стандартных устройств сопряжения с объектом (УСО), делающий их универсальными средствами управления для широкого круга применений. Тем не менее, на нижнем уровне управления двигателями, статическими преобразователями и др. требуется в большинстве случаев разработка встроенных специализированных контроллеров, причем эту разработку могут выполнить только специалисты в данной предметной области, хорошо разбирающиеся в особенностях объекта управления, способные предложить и реализовать весь комплекс необходимых алгоритмов оптимального управления объектом.

К устройствам сопряжения относится и оборудование для организации связи с удалёнными объектами. На сегодня уже разработаны системы, обеспечивающие надёжную связь, главным образом — беспроводную.

Главное отличие промышленного контроллера от промышленного компьютера состоит в адаптации языка программирования под конкретную область применения. Так, большинство промышленных контроллеров, выпускаемых такими известными фирмами как Alien Bradley, Simens, Fanuc, АВВ и др. являются по существу программируемыми логическими контроллерами (PLC) и имеют встроенные интерпретаторы с языка релейной автоматики или языка Булевой алгебры, что и позволяет неспециалисту программирование системы управления. Таким образом, одноуровневые системы встроенного управления выполняются преимущественно на базе однокристальных микро ЭВМ-микроконтроллеров. Двух- и более уровневые предполагают использование на среднем уровне управления одно- и много платных микро ЭВМ типа промышленного программируемого контроллера или компьютера в промышленном исполнении.

Встраиваемое управляющее устройство является микропроцессорной системой управления, в состав которой кроме центрального процессора на базе однокристального микропроцессора и микроконтроллера входят необходимые дополнительные элементы памяти и периферийные интерфейсные БИС для организации сопряжения с датчиками, объектом управления и системой управления более высокого уровня.

Основной базой для построения встроенных систем управления нижнего уровня являются именно однокристальные микро ЭВМ и микроконтроллеры, могут применяться также и законченные одноплатные системы управления на их основе, выпускаемые рядом фирм в качестве контроллеров-прототипов. Долгое время в нашей стране были доступны в основном только освоенные отечественной промышленностью младшие модели 8-разрядных микропроцессоров и микроконтроллеров фирмы Intel, совместимые с изделиями MCS-80, MCS-48, MCS-51, а также 16-разрядные процессоры собственной разработки, система команд которых совместимы процессорами фирмы DEC (1816BM1, 1816BM2).

В настоящее время ситуация резко изменилась, и вся самая передовая продукция ведущих фирм мира стала доступна отечественному разработчику и производителю.

Сегодня разработчики встроенных систем управления стоят перед непростым выбором: какое изделие и какой фирмы использовать в проекте.

Разработка мощной микропроцессорной системы управления является сложным делом. Для этого необходим штат высококвалифицированных инженеров схемотехников и программистов, а также соответствующее оборудование, а именно комплект аппаратных и программных средств разработки (кстати, для каждого набора микропроцессорных БИС - свой). Современные микропроцессоры и микроконтроллеры имеют высокие тактовые частоты и требуют исключительной тщательности как при проектировании печатной платы, включая разводку, так и при ее изготовлении.

Как известно в настоящее время одной из самых больших проблем в электронной промышленности является расширение разрыва в соотношении цена/характеристики между вентильными матрицами, программируемыми пользователем (ВМПП) и специализированными ИС (СпИС). Хотя ВМПП относительно недороги для разработки, они не обладают быстроедействием СпИС. И хотя СпИС хорошо применяются в системах с невысокими требованиями по мощности, однако единовременные затраты на их проектирование уже достигли уровня 20 млн. долларов. Всего несколько лет назад эти затраты составляли 100 тысяч долларов. Задачу изготовления быстроедействующих схем при допустимой стоимости при пониженных размерах партий поставили фирмы IBM, NEC и LSI Logic. В драку также вступает новичок, частная фирма Leopard Logic, основанная фирмами Acer Technology и Acorn Campus. В январе сего года Leopard Logic объявила о выпуске программируемых логических приборов Gladiator, которые президент фирмы К. Филипс назвал, как специализированные стандартные приборы, которые имеют быстроедействие в два, четыре раза превышающее быстроедействие ВМПП, а стоимость составляет одну десятую стоимости СпИС. В архитектуре используется комбинация логики, перестраиваемой с использованием масок и блоков, перестраиваемых с использованием СОЗУ. Структура, перестраиваемая с помощью масок, как и логическая структура структурированных СпИС, имеет импеданс межсоединений, подобный импедансу вентильных матриц. Типовая рабочая частота подсистем в структуре, перестраиваемой с помощью масок, составляет от 200 МГц до 300 МГц. Наличие на кристалле ВМПП структуры означает, что значительная часть конструкции может быть реализована в виде ВМПП.

Первый прибор серии содержит 42 К логических элементов с масочной перестройкой, 4К логических элемента с перестройкой от СОЗУ, встроенное СОЗУ емкостью более 2 Мбит и 64 умножителя-аккумулятора. Сложность этого прибора близка сложности современной старшей модели ВМПП. В серии Leopard этот прибор займет среднее положение. Схема содержит типовые перестраиваемые элементы В/В, но не поддерживает LVDS (низковольтные дифференциальные сигналы) и не имеет блоков параллельно-последовательных и последовательно-параллельных преобразователей.

Возможно самым важным архитектурным элементом является способ соединения структур. В данном случае используется сдвоенная структура сеток. Одна сетка обслуживает внутренние межсоединения, другая сетка реализует соединения между областями кристалла, но обе обеспечивают ресурсы доступа к любому логическому элементу. Таким образом, могут быть обеспечены сотни и тысячи межсоединений между структурой логики, программируемой пользователем и структурой логики, программируемой с помощью маски.

На микроархитектурном уровне конструкция схемы базируется на двух фундаментальных разработках ВМПП фирмы Leopard. Одна разработка представляет собой совмещение четырех входовой таблицы соответствия и триггера.

Другая разработка представляет собой схему межсоединений на базе мультиплексора.

Архитектура также соответствует последним достижениям в области структурированных СпИС. Так все пользовательские межсоединения объединены в один слой, так что здесь может быть использована одна заказная маска. Здесь также используется в качестве базового логического элемента таблица соответствия с программированием от СОЗУ, как это имеет место в ВМПП. Но имеется отличие от сетей вентилях или мультиплексоров, используемых в структурированных СпИС. Также в отличие от структурированной СпИС фирмы eASIC Corp., где каждый элемент может быть перепрограммирован, в архитектуре Leopard используются фиксированные блоки.

С точки зрения производителей микропроцессорной техники все задачи решаемые системами встроенного управления делятся на два больших класса: управление событиями в реальном времени и управление потоками данных. Каждый класс задач предъявляет свои специфические требования к микропроцессору или микроконтроллеру, что отражается, прежде всего, в наборе функций, реализуемых на кристалле, а также в системе команд.

К первому классу относятся задачи, требующие быстрой реакции микропроцессорной системы на изменение внешних условий (на срабатывание технологических датчиков, изменение параметров и т.д.). Как правило, системы управления приводами, энергетическими установками, роботами, а также системы распределенной автоматизации относятся к системам первого класса. Эти задачи требуют применения микроконтроллеров с большим объемом интегрированной на кристалл периферии, включая реализацию на кристалле памяти программ, памяти данных и устройств ввода-вывода, что сокращает аппаратные затраты и удешевляет изделие со встроенной системой управления. Чаще всего в системах управления этого класса для реализации алгоритма управления требуется память относительно небольшого объема (до 32 Кбайт).

Ко второму классу задач относятся задачи, требующие быстрой обработки значительных объемов информации. Например, в микропроцессорных системах поддержки компьютерных сетей, в системах

управления летательными аппаратами, подвижным составом, в системах обработки видео изображений, когда встроенный процессор должен выполнять множество различных вычислительных операций, в том числе операций с плавающей запятой. Как правило, для решения таких задач требуется уже высокопроизводительный 32- или 64-разрядный процессор. В соответствии с упомянутыми выше классами задач, встроенные микропроцессорные контроллеры могут быть разделены на следующие группы:

- 8-битовые микроконтроллеры первого
- Современные 8-разрядные микроконтроллеры (MCSR-51, MCSR-51, MCSR-251).
- 16-разрядные микроконтроллеры для управления в реальном времени

Широкое распространение персональных компьютеров и микрокомпьютерных рабочих станций в бизнесе началось в начале 80-х годов. В настоящее время это оборудование представляет собой столь же неотъемлемую принадлежность любого офиса, как, например, телефон. До относительно недавних пор офисные локальные сети предоставляли в основном услуги соединения — они соединяли персональные компьютеры и терминалы с мэйнфреймами и системами промежуточного звена, на которых работали корпоративные приложения, а также обеспечивали соединение для рабочих групп на уровне отделов или цехов. В обоих случаях объемы передаваемых по сетям данных были невелики. Трафик в основном состоял из файлов и электронной почты. Локальные сети, предназначенные для обслуживания подобного трафика, а это в первую очередь Ethernet и Token Ring, отлично подходят для такого окружения.

В 90-е годы роль персонального компьютера (а вместе с ней и требования к локальной сети) изменилась благодаря двум важным тенденциям:

- Вычислительная мощность персональных компьютеров продолжает радовать экспоненциальным ростом. Все более мощные платформы поддерживают приложения, интенсивно использующие графику, и все усложняющиеся графические интерфейсы пользователя с операционными системами.

- Организации, занимающиеся административными информационными системами, увидели в локальных сетях жизнеспособные и важные вычислительные платформы. Начало этой тенденции положили системы «клиент-сервер», ставшие доминирующей архитектурой в деловой среде, а также более недавнее веяние — технология интранет.

Результатом этих тенденций стала необходимость обеспечения роста передаваемого по локальным сетям объема данных и, так как приложения стали более интерактивными, уменьшения времени задержки при передаче данных. Первые поколения сетей Ethernet, работавшие со скоростью 10 Мбит/с, и сетей Token ring со скоростью 16 Мбит/с просто не могли соответствовать новым требованиям.

Ниже перечислены примеры, для которых могут потребоваться высокоскоростные сети:

- Централизованные серверные фермы. Во многих приложениях необходимо поддерживать возможность загрузки больших объемов данных пользователем или клиентом с многочисленных централизованных серверов, называемых серверными фермами (server farms). В качестве примера можно назвать издательскую работу с цветными изображениями, когда серверы содержат десятки гигабайтов изображений, загружаемых на рабочие станции. С тех пор как производительность серверов увеличилась, узким местом стала сеть. Эта проблема не может быть решена при помощи коммутируемой сети Ethernet, так как пропускная способность каждой линии связи с клиентом ограничена значением 10 Мбит/с.

- Мощные рабочие группы. Эти группы, как правило, состоят из небольшого числа сотрудничающих пользователей, которым необходимо перемещать по сети массивные файлы данных. Например, это может быть группа разработчиков программного обеспечения, тестирующих новую версию программы, или компания, занимающаяся автоматизированным проектированием (Computer-Aided Design, CAD) и регулярно запускающая свои новые модели. В таких случаях большие объемы данных распределены по нескольким рабочим станциям. Они обрабатываются и обновляются на очень высоких скоростях, так как процесс формирования модели требует множества итераций.

- Высокоскоростная локальная магистраль. По мере роста потребности в обработке данных локальных сетей становится все больше, что требует высокоскоростного взаимодействия.

Ключ к проектированию высокопроизводительных сетей заключается в способности моделировать и оценивать параметры производительности. Разработчик должен быть способен на основании наблюдений оценить объем и характеристики будущего трафика. Статистические характеристики трафика влияют на разнообразные аспекты проектирования и конфигурирования, включая протоколы маршрутизации, протоколы резервирования ресурсов, дисциплины очередей в маршрутизаторах и АТМ-коммутаторах, а также размеры буферов. Более того, пользователь должен уметь охарактеризовать планируемый трафик, чтобы принять верные решения в области резервирования ресурсов.

Для описания потока данных большое значение представляет ряд параметров.

- ♦ Характеристики пропускной способности:

- ♦ Средняя скорость. Средняя нагрузка на сеть, оказываемая источником, представляет собой ключевой параметр в определении объема ресурсов, которые должны быть выделены этому источнику. Средняя скорость передачи данных определяет тот поток, который источник может поддерживать в течение длительного периода времени.

♦ Пиковая скорость. Этот параметр определяет для сети максимальный трафик, который она в состоянии поддерживать, либо выделяя соответствующие ресурсы, либо резервируя достаточный объем буферного пространства для сглаживания пульсаций.

♦ Неравномерность. Пиковая скорость представляет собой один из критериев неравномерности. Более точным критерием является неравномерность пропускной способности. Неравномерность характеризует непостоянство трафика источника и представляет собой индикатор того, до какой степени статистическое мультимплексирование может использоваться для повышения эффективности.

♦ Характеристики задержки:

♦ Задержка передачи. Этот параметр представляет собой задержку, вносимую сетью при передаче данных от отправителя к получателю. Максимальное время задержки также может быть требованием, которое выдвигается приложением.

♦ Вариация задержки. Вариация задержки является важным параметром приложений реального времени, в которых данные, воспроизводящиеся получателем, должны прибывать с постоянной скоростью, равной скорости, с которой отправитель эти данные передает.

Эти и подобные им параметры очень важны для конфигурации сети и устройства протоколов. Для принятия эффективных решений необходимо довольно точно моделировать трафик данных.

Анализ очередей представляет собой простое и легко интерпретируемое средство получения полезных результатов для управления работой по проектированию сетей. В течение десятилетий анализ очередей основывался на предположении о соответствии типа трафика распределению Пуассона. Затем совершенно неожиданно исследователи из Бостонского университета и лаборатории Bellcore опубликовали удивительные результаты. Как оказалось, по крайней мере в некоторых случаях трафик описывается не распределением Пуассона, а является по своей природе самоподобным, или фрактальным. При таком трафике производительность сети не подчиняется аккуратным формулам анализа очередей, а имеют место большие задержки и снижение пропускной способности. С момента публикации эти результаты были многократно подтверждены на трафиках самых разных типов

Глава 2

Анализ качества обслуживания сети пользователей

2.1 Основные понятия методологического аппарата

2.1.1 Случайные переменные

Случайная переменная (random variable) представляет собой отображение множества всех возможных событий в пространстве выборок на пространство вещественных чисел. Таким образом, случайная переменная связывает с каждым событием вещественное число. Эта концепция иногда выражается в виде эксперимента с множеством возможных результатов. При этом случайная переменная присваивает значение каждому результату. Таким образом, значение случайной переменной является случайной величиной. Мы дадим следующее формальное определение. Случайная переменная X представляет собой функцию, ставящую в соответствие каждому результату в пространстве выборок число, удовлетворяющую следующим условиям:

Множество $\{X < x\}$ является событием для каждого значения x .

$$\Pr[X = -\infty] = \Pr[X = +\infty] = 0.$$

Случайная переменная является непрерывной (continuous), если она может принимать бесконечное множество различных значений. Случайная переменная является дискретной (discrete), если она может принимать конечное или ограниченное множество различных значений.

Функции распределения и плотности

Непрерывная случайная переменная X может быть описана либо с помощью функции распределения (distribution function) $F(x)$, либо с помощью функции плотности (density function) $f(x)$:

♦ функция распределения:

$$F(x) = \Pr[X \leq x] \quad F(-\infty) = 0 \\ F(+\infty) = 1.$$

♦ функция плотности:

$$f(x) = \frac{d}{dx} F(x);$$

$$F(x) = \int_{-\infty}^x f(y) dy;$$

$$\int_{-\infty}^{+\infty} f(y) dy = 1.$$

Для дискретной случайной переменной ее распределение вероятности характеризуется следующими уравнениями:

$$P_X(k) = \Pr[X = k];$$

$$\sum_{\text{Все } k} P_X(k) = 1.$$

Непрерывная случайная переменная X может быть описана либо с помощью функции распределения (distribution function) $F(x)$, либо с помощью функции плотности (density function) $f(x)$:

Часто нас интересует не все распределение, а какая-нибудь одна характеристика случайной переменной, например:

♦ Среднее значение (также называемое ожидаемым значением, или первым моментом)

$$\begin{cases} E[X] = \mu_X = \int_{-\infty}^{\infty} xf(x)dx & \text{— непрерывный случай;} \\ E[X] = \mu_X = \sum_{\text{Все } k} k \Pr[x = k] & \text{— дискретный случай.} \end{cases}$$

♦ Второй момент:

$$\begin{cases} E[X^2] = \mu_{X^2} = \int_{-\infty}^{\infty} x^2 f(x)dx & \text{— непрерывный случай} \\ E[X^2] = \mu_{X^2} = \sum_{\text{Все } k} k^2 \Pr[x = k] & \text{— дискретный случай} \end{cases}$$

♦ Дисперсия: $\text{Var}[X] = E[(X - \mu_X)^2] = E[X^2] - \mu_X^2$.

♦ Среднеквадратичное отклонение: $\sigma_X = \sqrt{\text{Var}[X]}$.

Дисперсия (variance) и среднеквадратичное отклонение (standard deviation) являются показателями рассеяния значений вокруг средней величины. Высокое значение дисперсии переменной означает, что переменная чаще и дальше отклоняется от среднего значения, чем переменная с низким значением дисперсии. Несложно показать, что для любой постоянной a справедливы следующие уравнения:

$$E[aX] = aE[X]$$

$$\text{Var}[aX] = a^2 \text{Var}[X].$$

Среднее значение также называют статистикой первого порядка; второй момент и дисперсия представляют собой статистику второго порядка. Из функции плотности вероятности также можно вывести статистики более высоких порядков.

Важные распределения

В данном разделе будут описаны некоторые распределения, играющие важную роль в теории анализа очередей.

2.1.2 Экспоненциальное распределение

$$F(x) = 1 - e^{-\lambda x};$$

$$f(x) = \lambda e^{-\lambda x};$$

$$x \geq 0.$$

Интересное свойство экспоненциального распределения состоит в том, что его среднее значение равно среднеквадратичному отклонению:

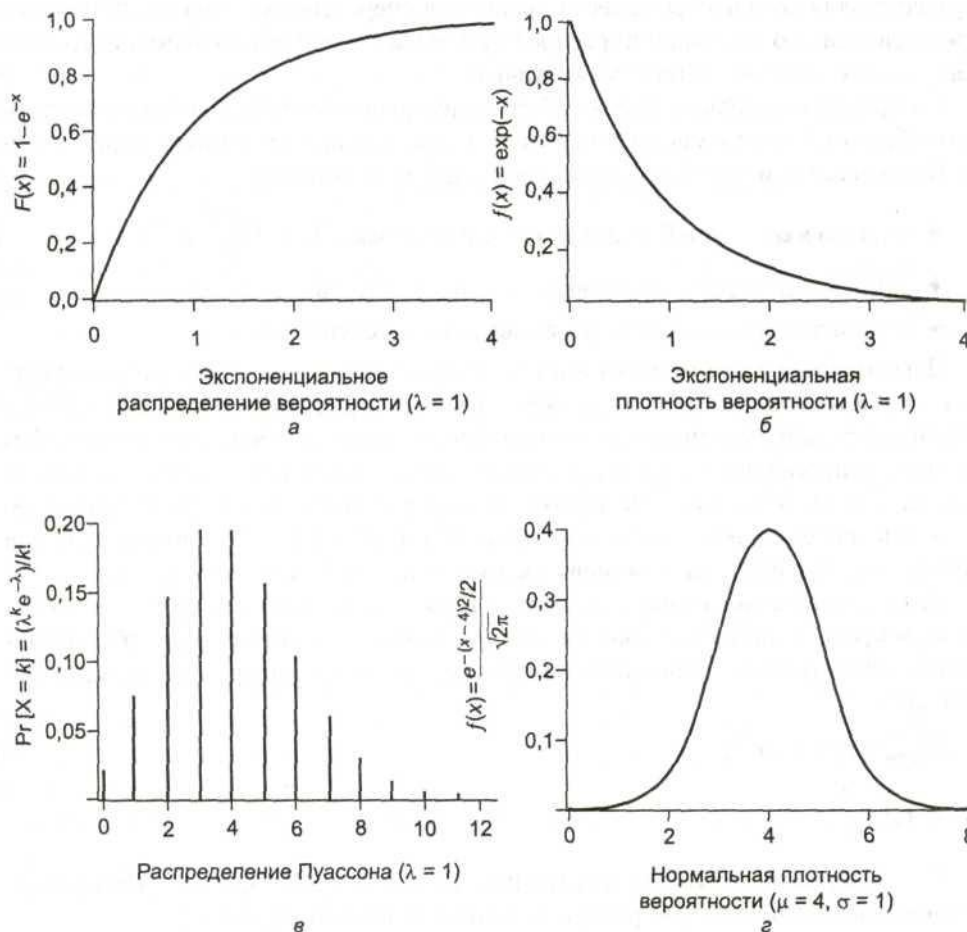
$$E[x] = \sigma_x = \frac{1}{\lambda}.$$

Экспоненциальное распределение с параметром $\lambda > 0$ описывается следующими функциями распределения и плотности:

В случае применения к интервалу времени, например времени обслуживания, такое распределение иногда называют случайным распределением. Это связано с тем, что если интервал времени уже начался,

то он может закончиться в произвольный момент времени с равной вероятностью.

Важность этого распределения в теории очередей связана с тем, что мы часто можем предположить, что время обслуживания распределено экспоненциально. В случае телефонного трафика время обслуживания — это время, в течение которого абонент занимает оборудование. В сети с коммутацией пакетов время обслуживания представляет собой время передачи данных, и поэтому оно пропорционально длине пакета. Трудно дать четкое теоретическое обоснование тому, почему время обслуживания должно иметь экспоненциальное распределение, но во многих случаях его распределение близко к экспоненциальному. И это хорошо, так как значительно упрощает анализ очередей.



2.1.3 Распределение Пуассона

Другое важное распределение — распределение Пуассона с параметром $X > 0$, принимающее значения в точках $0, 1, \dots$:

$$\Pr[X = k] = \frac{\lambda^k}{k!} e^{-\lambda}; \quad k = 0, 1, 2, \dots;$$

$$E[X] = \text{Var}[X] = \lambda.$$

Если $X < 1$, тогда $\Pr[X=k]$ достигает максимума при $k = 0$. Если $X > 1$, но не является целым числом, тогда $\Pr[X=k]$ достигает максимума при k , равном

наибольшему целому числу, меньшему, чем X . Если X представляет собой положительное целое число, тогда у функции распределения Пуассона два максимума при $k = X$ и $k = X - 1$.

Распределение Пуассона также представляет важность для анализа очередей, так как мы должны предположить, что частота прибытия пакетов распределена по Пуассону, для того чтобы вывести уравнения очередей. К счастью, предположение о подчинении распределения частоты прибывающих пакетов закону Пуассона оказывается справедливым.

Распределение Пуассона может быть применено к частоте получения следующим образом. Если элементы поступают в очередь в соответствии с распределением Пуассона, это может быть выражено следующим образом:

- ♦ $\Pr[k \text{ элементов прибывают в течение интервала } T] = \frac{(\lambda T)^k}{k!} e^{-\lambda T}$;
- ♦ $E[\text{число элементов, прибывших в течение интервала } T] = \lambda T$;
- ♦ средняя скорость прибытия, в элементах в секунду $= \lambda$.

Пакеты, прибывающие в соответствии с законом Пуассона, часто называют случайно прибывающими. Это объясняется тем, что вероятность прибытия пакета в течение короткого интервала времени пропорциональна длительности интервала времени и не зависит от времени, прошедшего с момента прибытия последнего пакета. Таким образом, если прибытие пакетов подчиняется распределению Пуассона, пакет с равной вероятностью может прибыть в любой момент времени независимо от того, в какие моменты времени прибывают другие пакеты.

Другое интересное свойство процесса Пуассона связано с его экспоненциальным распределением. Оказывается, что интервалы времени между прибывающими в соответствии с распределением Пуассона пакетами распределены экспоненциально:

$$\Pr[T_a < t] = 1 - e^{-\lambda t};$$

$$E[T_a] = \frac{1}{\lambda}.$$

Таким образом, как и следовало ожидать, среднее время между прибытием пакетов представляет собой обратную величину от частоты прибытия.

2.1.4 Нормальное распределение

Нормальное распределение имеет следующие функцию плотности и функцию распределения:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2};$$

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-(y-\mu)^2 / 2\sigma^2} dy.$$

Причем

$$E[X] = \mu;$$

$$\text{Var}[X] = \sigma^2.$$

Большое значение представляет центральная предельная теорема, утверждающая, что средняя величина от большого количества независимых случайных переменных будет распределена приблизительно нормально и почти не зависит от их индивидуальных распределений. Одно ключевое требование заключается в конечности среднего значения и дисперсии. Центральная предельная теорема играет в статистике ключевую роль.

2.1.5 Стохастические процессы

Стохастический процесс (stochastic process), также называемый случайным процессом (random process), представляет собой семейство случайных переменных $(x(t), t \in T)$, индексированных параметром t на некотором множестве индексов T . Как правило, множество индексов интерпретируется как измерение времени, а $x(t)$ является функцией времени. Другими словами, стохастический процесс представляет собой случайную переменную, являющуюся функцией времени. Непрерывный во времени стохастический процесс (continuous-time stochastic process) — это стохастический процесс, в котором значение времени t изменяется непрерывно, как правило, вдоль оси неотрицательных вещественных чисел $\{x(t), 0 < t < \Gamma\}$, хотя иногда оно может изменяться вдоль всей оси вещественных чисел. Дискретный во времени стохастический процесс (discrete-time stochastic process) — это стохастический процесс, в котором значение времени t принимает дискретные, как правило, положительные целые значения $(x(t), t = 1, 2, \dots)$, хотя иногда оно может принимать и целые значения во всем интервале от $-\infty$ до $+\infty$.

Вспомним, что случайная переменная определяется как функция, преобразующая результат эксперимента в некоторое значение. Учитывая это, выражение для $x(t)$ может быть интерпретировано по-разному:

Как семейство функций времени (переменная t ; все возможные результаты).

Как отдельная функция времени (переменная t ; один результат).

Как случайная переменная (переменная t фиксированная; все возможные результаты).

Как отдельное число (переменная t фиксированная; один результат).

Конкретная интерпретация $x(t)$ обычно ясна из контекста.

Следует сказать несколько слов о терминологии. Непрерывный по значениям стохастический процесс (continuous-value stochastic process) — это стохастический процесс, в котором случайная переменная $x(t)$ с фиксированным t (случай 3) принимает непрерывные значения. Дискретный по значениям стохастический процесс (discrete-value stochastic process) — это стохастический процесс, в котором случайная переменная в любой момент t принимает конечное или счетно-бесконечное количество значений. И непрерывный, и дискретный во времени стохастические процессы могут быть как непрерывными, так и дискретными по значениям.

Как для любой случайной переменной, функцию $x(t)$ с фиксированным значением t можно охарактеризовать функциями распределения

вероятностей и плотности вероятности. Для непрерывного по значениям стохастического процесса эти функции принимают следующий вид:

♦ *функция распределения:*

$$F(x; t) = \Pr[\mathbf{x}(t) \leq x];$$

$$F(-\infty; t) = 0;$$

$$F(\infty; t) = 1;$$

♦ *функция плотности:*

$$f(x; t) = \frac{\partial}{\partial x} F(x; t);$$

$$F(x; t) = \int_{-\infty}^x f(y; t) dy;$$

$$\int_{-\infty}^{\infty} f(y; t) dy = 1.$$

$$P_{\mathbf{x}(t)}(k) = \Pr[\mathbf{x}(t) = k].$$

$$\sum_{\text{Все } k} P_{\mathbf{x}(t)}(k) = 1.$$

Для дискретного по значениям стохастического процесса:

Полная статистическая характеристика стохастического процесса должна учитывать переменную времени. Используя первый вариант интерпретации из предыдущего списка, стохастический процесс $\mathbf{x}(t)$ включает в себя бесконечное количество случайных переменных, по одной для каждого значения t . Чтобы полностью описать статистику процесса, нам потребуется указать функцию совместной плотности вероятности переменных $\mathbf{x}(t_1)$, $\mathbf{x}(t_2)$, ..., $\mathbf{x}(t_n)$ для всех значений n ($1 < n < \infty$) и всех возможных отсчетов времени (t_1, t_2, \dots, t_n). Но для задач нашей книги нет необходимости углубляться в эту тему.

2.2 Анализ очередей

В области обмена данными и компьютерных сетей вам часто может понадобиться умение предсказывать результат определенных изменений в системе, таких как увеличение нагрузки или доработка конструкции. Например, организация поддерживает определенное количество терминалов, персональных компьютеров и рабочих станций, подключенных к 100-мегабитной локальной сети. Предполагается подключение к локальной сети дополнительного отдела в здании. Сможет ли существующая локальная сеть выдержать увеличение нагрузки или лучше создать вторую локальную сеть и соединить ее с первой сетью мостом?

Бывают другие случаи, в которых сеть еще не существует, но на основе имеющихся требований нужно создать проект сети. Например, руководство собирается установить в некоем отделе персональные компьютеры и создать из них локальную сеть с файловым сервером. Основываясь на результатах эксперимента в другом отделе компании, вы можете оценить нагрузку, создаваемую каждым персональным компьютером, и вычислить требуемые мощности локальной сети и файлового сервера.

В каждом случае основной заботой является производительность системы. В интерактивном приложении или в приложении реального времени это, как правило, означает время отклика. В других случаях главным вопросом является пропускная способность.

Для оценки производительности проектируемой системы требуется определенный механизм предсказаний. Решение сетевых проблем и проблем связи, а также многих других проблем из реального мира часто можно получить с помощью аналитических моделей, основанных на теории очередей.

Количество вопросов, которые можно решить путем анализа очередей, бесконечно. Круг этих вопросов охватывает практически все темы, обсуждаемые в данной книге. Таким образом, умение применять анализ крайне важно для разработчиков, занимающихся любой из этих проблем.

Хотя математическая часть теории очередей сложна, применение самой теории в целях анализа производительности во многих случаях оказывается довольно простым. Все, что для этого требуется, — это знание элементарных статистических понятий (средних значений и дисперсий), а также понимание в общих чертах методов применения теории очередей. Вооружившись этими знаниями, разработчик часто может выполнить расчеты на клочке бумаги, используя несложные таблицы очередей, или с помощью простой компьютерной программы, занимающей всего несколько строк кода.

В этой главе предлагается практический курс анализа очередей.

Простой пример поведения очередей

Прежде чем перейти к деталям анализа очередей, рассмотрим грубый пример, чтобы «почувствовать» тему данной главы. Рассмотрим веб-сервер, способный обслуживать индивидуальные запросы в среднем за 1 мс. Чтобы

упростить задачу, предположим, что сервер обслуживает каждый запрос ровно за 1 мс. Если скорость поступления запросов равна одному запросу в миллисекунду (1000 запросов в секунду), то, как кажется, можно утверждать, что сервер справится с такой нагрузкой.

Предположим, что запросы прибывают с постоянной скоростью с интервалом ровно в 1 мс. Сервер обрабатывает запрос сразу, как только он приходит. Как только сервер завершает обработку текущего запроса, поступает новый запрос, и сервер снова принимается за работу.

Теперь рассмотрим более реалистичный вариант, в котором средняя скорость поступления запросов составляет также тысячу запросов в секунду, но эта скорость не постоянна. В течение любого заданного миллисекундного интервала может не поступить ни одного запроса, поступить один запрос или поступить несколько запросов, но в среднем за каждую миллисекунду приходит один запрос. В этом случае здравый смысл подсказывает, что сервер справится. В периоды высокой нагрузки, когда поступает больше запросов, чем сервер способен обработать, сервер может хранить избыточные запросы в буфере. Можно сказать, что поступающие запросы ставятся в очередь на обработку. В спокойные периоды сервер может на-верстать упущенное и очистить буфер. В данном случае важный вопрос проектирования заключается в том, насколько большим должен быть буфер.

Таблицы 2.1-2.3 дают грубое представление о поведении такой системы. В табл. 2.1 мы предполагаем, что в систему поступает в среднем 500 запросов в секунду, что соответствует половине производительности сервера. Записи в таблице показывают количество запросов, прибывающих каждую секунду, количество запросов, обработанных в течение этой секунды, и количество избыточных запросов, поставленных в очередь к концу секундного интервала. В таблице содержатся данные о состоянии очереди на протяжении 50 с. Среднее по 50-секундным интервалам значение числа запросов, стоящих в очереди, равняется 43. Пиковое значение числа запросов, стоящих в очереди, составляет 600. В табл. 2.1 средняя скорость поступления запросов увеличена до 95 % от производительности сервера и составляет 950 запросов в секунду. При этом среднее количество запросов, ожидающих обработки в буфере, увеличивается до 1859. Это может показаться удивительным. Скорость поступления запросов увеличилась менее чем в два раза, тогда как средняя длина очереди выросла в 40 раз. В табл. 2.3 средняя скорость поступления запросов еще немного увеличена — до 99 % от производительности сервера, — в результате среднее количество запросов в буфере выросло до 2583. Таким образом, совсем незначительное увеличение средней скорости поступления запросов приводит к почти 40-процентному росту числа запросов, ожидающих в очереди.

**Таблица 2.1. Поведение очереди при нормализованной скорости
прибытия, равной 0,5**

Время	Вход	Выход	Очередь
0	0	0	0
1	88	88	0
2	796	796	0
3	1627	1000	627
4	51	678	0
5	34	34	0
6	966	966	0
7	714	714	0
8	1276	1000	276
9	494	769	0
10	933	933	0
11	107	107	0
12	241	241	0
13	16	16	0
14	671	671	0
15	643	643	0
16	812	812	0
17	262	262	0
18	218	218	0
19	1378	1000	378
20	507	885	0
21	15	15	0
22	820	820	0
23	1253	1000	253
24	307	559	0
25	540	540	0
26	190	190	0
27	500	500	0
28	96	96	0
29	943	943	0
30	105	105	0
31	183	183	0
32	447	447	0
33	542	542	0
34	166	166	0
35	165	165	0
36	490	490	0
37	510	510	0
38	877	877	0
39	37	37	0
40	163	163	0
41	104	104	0
42	42	42	0
43	291	291	0
44	645	645	0
45	363	363	0
46	134	134	0
47	920	920	0
48	1507	1000	507
49	598	1000	105
50	172	277	0
Среднее	499	499	43

**Таблица 2.2. Поведение очереди при нормализованной скорости
прибытия, равной 0,95**

Время	Вход	Выход	Очередь
0	167	167	0
2	1512	1000	512
3	3091	1000	2604
4	97	1000	1701
5	65	1000	765
6	1835	1000	1601
7	1357	1000	1957
8	2424	1000	3382
9	939	1000	3320
10	1773	1000	4093
11	203	1000	3269
12	458	1000	2754
13	30	1000	1784
14	1275	1000	2059
15	1222	1000	2281
16	1543	1000	2824
17	498	1000	2322
18	414	1000	1736
19	2618	1000	3354
20	963	1000	3317
21	29	1000	2346
22	1558	1000	2904
23	2381	1000	4285
24	583	1000	3868
25	1026	1000	3894
26	361	1000	3255
27	950	1000	3205
28	182	1000	2387
29	1792	1000	3179
30	200	1000	2378
31	348	1000	1726
32	849	1000	1575
33	1030	1000	1605
34	315	1000	921
35	314	1000	234
36	931	1000	165
37	969	1000	134
38	1666	1000	800
39	70	871	0
40	310	310	0
41	198	198	0
42	80	80	0
43	553	553	0
44	1226	1000	226
45	690	915	0
46	255	255	0
47	1748	1000	748
48	2863	1000	2611
49	1136	1000	2748
50	327	1000	2074
Среднее	948	907	1859

**Таблица 2.3. Поведение очереди при нормализованной скорости
прибытия, равной 0,99**

Время	Вход	Выход	Очередь
0	0	0	0
1	174	174	0
2	1576	1000	576
3	3221	1000	2798
4	101	1000	1899
5	67	1000	966
6	1913	1000	1879
7	1414	1000	2292
8	2526	1000	3819
9	978	1000	3797
10	1847	1000	4644
11	212	1000	3856
12	477	1000	3333
13	32	1000	2365
14	1329	1000	2693
15	1273	1000	2967
16	1608	1000	3574
17	519	1000	3093
18	432	1000	2525
19	2728	1000	4253
20	1004	1000	4257
21	30	1000	3287
22	1624	1000	3910
23	2481	1000	5391
24	608	1000	4999
25	1069	1000	5068
26	376	1000	4445
27	990	1000	4435
28	190	1000	3625
29	1867	1000	4492
30	208	1000	3700
31	362	1000	3062
32	885	1000	2947
33	1073	1000	3020
34	329	1000	2349
35	327	1000	1676
36	970	1000	1646
37	1010	1000	1656
38	1736	1000	2392
39	73	1000	1465
40	323	1000	788
41	206	994	0
42	83	83	0
43	576	576	0
44	1277	1000	277
45	719	996	0
46	265	265	0
47	1822	1000	822
48	2984	1000	2805
49	1184	1000	2990
50	341	1000	2330
Среднее	988	942	2583

2.3 Моделирование очередей к серверам

Цели анализа очередей

Часто бывает необходимо рассчитать требуемую производительность на основе имеющейся информации о нагрузке или на основе оценки нагрузки в новом окружении. Для этого могут применяться различные подходы:

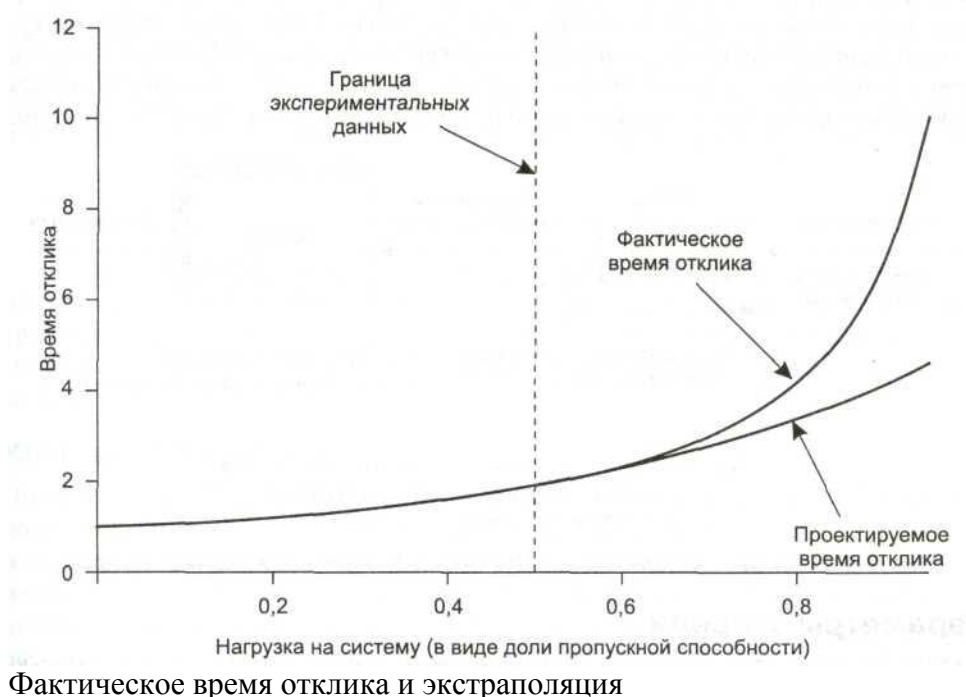
- ♦ Выполнение анализа на уже готовом оборудовании на основе фактических значений.
- ♦ Выполнение простых прогностических расчетов на основе экстраполяции данных, полученных на существующем оборудовании.
- ♦ Разработка аналитической модели на основе теории очередей.
- ♦ Создание и запуск программной модели.

Первый вариант вообще не является вариантом. Мы просто собираем систему наугад и смотрим, что получится. Такой подход приведет лишь к появлению недовольных пользователей и к неразумному расходованию средств. Второй вариант кажется более многообещающим. Аналитик может заявить, что спрогнозировать будущие требования с какой-либо степенью точности невозможно. Поэтому нет смысла заниматься точным моделированием. Вместо этого простой и грубый прогноз позволит получить приближенную оценку. Недостаток такого подхода состоит в том, что поведение большинства систем при изменении нагрузки оказывается довольно неожиданным, как было показано в разделе 8.1. Если имеется окружение, в котором есть коллективно используемый ресурс (например, сеть, линия передачи данных или система разделения времени), тогда производительность такой системы, как правило, отвечает на увеличение нагрузки экспоненциальным увеличением времени отклика.

На рисунке показан соответствующий пример. Верхняя линия демонстрирует увеличение времени отклика на запрос пользователя к общему ресурсу при увеличении нагрузки на этот ресурс. Нагрузка измеряется в долях от максимальной производительности ресурса. Таким образом, если мы имеем дело с маршрутизатором, способным обрабатывать и переправлять 1000 пакетов в секунду, то нагрузка 0,5 соответствует средней скорости поступления 500 пакетов в секунду. Время отклика представляет собой время, требуемое для пересылки любого входящего пакета. Нижняя линия иллюстрирует экстраполяцию многочленом третьего порядка экспериментального участка кривой времени отклика, соответствующего нагрузке от 0 до 0,5 мощности ресурса. Обратите внимание на то, что данный способ экстраполяции работает более или менее удовлетворительно лишь на участке от 0,5 до 0,7 от мощности ресурса, а когда нагрузка начинает «зашкаливать» за 0,8 или 0,9, наступает крах системы.

Таким образом, требуется более точный метод предсказаний. Третий вариант заключается в использовании аналитической модели, представляющей собой набор уравнений, которые могут быть решены для получения требуемых параметров (времени отклика, пропускной

способности и т. д.). Аналитические модели, основанные на теории очередей, обеспечивают довольно хорошее соответствие с реальностью в таких областях, как компьютеры, операционные системы и сети. Недостаток теории очередей заключается в том, что для составления уравнений относительно интересующих нас параметров приходится принимать ряд упрощающих допущений.

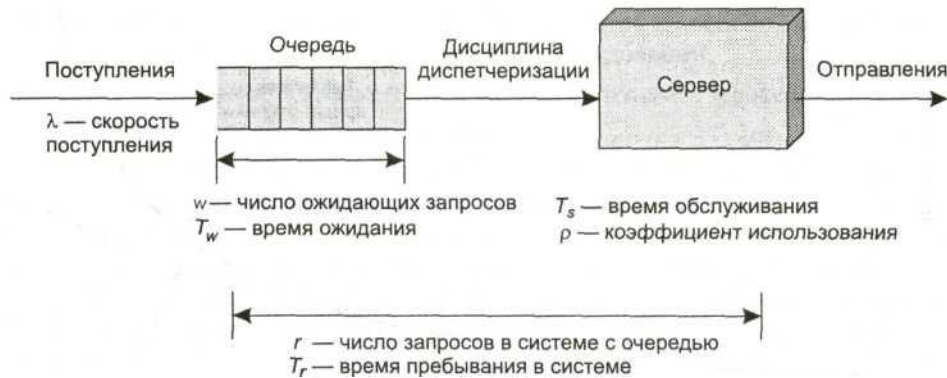


Последний подход представляет собой численное моделирование. В данном случае при наличии достаточно мощного и гибкого специализированного языка программирования аналитик может очень подробно смоделировать реальную систему, не прибегая к многочисленным допущениям, требуемым в теории очередей. Однако в большинстве случаев численная модель не требуется или, по меньшей мере, не рекомендуется на первом этапе анализа. Во-первых, и измерения существующей системы, и прогностические оценки будущей нагрузки содержат определенные погрешности. Таким образом, независимо от того, насколько хорошей будет модель, качество результатов ограничено качеством исходных данных. Во-вторых, несмотря на то что в теории очередей требуется принять множество допущений, получаемые результаты часто оказываются довольно близкими к тем, которые могут быть получены при более тщательном численном моделировании. Более того, для четко поставленной задачи анализ очередей может быть выполнен буквально за считанные минуты, тогда как создание, отладка и прогон моделирующей программы могут занять дни, недели и месяцы.

Именно по этим причинам аналитик обязан знать основы теории очередей.

Модели очередей **Очередь к одному серверу**

Простейшая система с очередью показана на рис. 8.2. Центральный элемент системы представляет собой сервер, предоставляющий другим элементам системы некоторые услуги. На сервер поступают запросы на обслуживание. Если сервер ничем не занят, запрос обрабатывается незамедлительно. В противном случае поступивший запрос помещается в очередь. Когда сервер завершает обработку запроса, пакет с обработанным запросом покидает сервер. Если на этот момент в очереди имеются необработанные запросы, один из них немедленно выбирается сервером.



Структура и параметры системы очередей для очереди с одним сервером

Параметры очереди

На рисунке также показаны некоторые важные параметры, связанные с моделью очередей. Запросы прибывают на обрабатывающее устройство с некоей средней скоростью X (заказов в секунду). Среди примеров поступающих заказов можно назвать пакеты, прибывающие на маршрутизатор, или звонки, поступающие на телефонный коммутатор. В любой заданный момент времени определенное количество заказов (ноль или больше) будет ожидать в очереди. Среднее количество ожидающих заказов равно w , а среднее время ожидания в очереди — T_w . Время T_w усредняется по всем поступающим заказам, включая те, которые обрабатываются без ожидания. Сервер обрабатывает поступающие заказы за среднее время обслуживания T_s . Это интервал времени от начала обработки заказа сервером до того момента, когда обработанный пакет покидает сервер. Коэффициент использования сервера ρ означает долю времени, которую сервер занят работой. Наконец, два параметра применяются ко всей системе в целом. Это r — среднее количество заказов в системе, включая обслуживаемый в данный момент заказ, плюс заказы, ждущие своей очереди, а также T_r — среднее время, которое заказ проводит в системе, включая ожидание в очереди и обработку, называемое средним временем пребывания в системе (mean residence time).

Если предположить, что емкость очереди бесконечна, тогда система никогда не теряет заказов. Их обслуживание просто откладывается на более позднее время. При увеличении скорости поступления заказов увеличивается коэффициент загрузки, а вместе с ним вероятность перегрузки. Очередь

становится длиннее, время ожидания в очереди возрастает. При $\rho = 1$ наступает насыщение, то есть сервер работает 100 % времени. Пока коэффициент загрузки меньше 100 %, сервер справляется с обработкой поступающих заказов, таким образом, средняя скорость на выходе сервера равняется средней скорости на входе. Как только сервер насыщается, скорость на выходе сервера остается постоянной независимо от того, насколько большей является скорость на входе. Таким образом, теоретический максимум входной скорости, при которой система будет успевать обрабатывать все запросы, равен:

$$\lambda_{\max} = \frac{1}{T_s}.$$

Однако при приближении к точке насыщения очереди становятся очень длинными и при $\rho = 1$ неограниченно растут. Из практических соображений, таких как требования ко времени отклика или ограничения на размеры буферов, входная скорость, как правило, ограничивается уровнем 70-90 % от теоретического максимума.

Ключевые моменты

Полезно представить процессы, связанные с очередями, на примере. На рисунке показан пример реализации процесса функционирования очереди. На графике изображена зависимость общего количества запросов в системе от времени. Затененные области обозначают периоды времени, когда сервер занят. На оси времени отмечены два типа событий: поступление запроса i в момент времени A_i и завершение обслуживания запроса i в момент времени D_i . Время, которое запрос i проводит в системе, равно $TR_i = D_i - A_i$. Фактическое время обслуживания запроса i обозначается как TS_i .

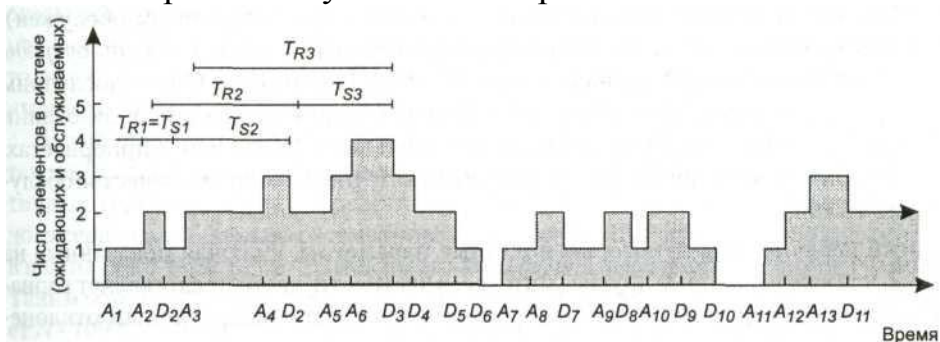


Иллюстрация процесса функционирования очереди

В данном примере время $Tr1$, которое запрос 1 проводит в системе, полностью состоит из времени обслуживания $Ts1$, так как когда он прибывает в систему, система пуста и поэтому может незамедлительно приступить к обслуживанию запроса. Время $TR2$ состоит из времени, в течение которого запрос 2 ждет обслуживания $\{D \setminus - A2\}$, и его времени обслуживания $Ts2$. Соответственно, $TR3 = (D3 - A3) = (D3 - D2) + (D2 - A3) = Ts3 + (D2 - A3)$. Однако запрос n может покинуть систему, прежде чем поступит запрос $n + 1$ (например, $D_n < A_{n+1}$), поэтому общая формула выглядит как $Tr_{n+1} = Ts_{n+1} + \text{MAX}[0, D_n - A_{n+1}]$.

Характеристики модели

Прежде чем выводить любые аналитические уравнения для модели очередей, следует выбрать определенные ключевые характеристики модели. Ниже приводятся типичные характеристики, как правило, применяемые в контексте передачи данных.

♦ Совокупность запросов (item population). Предполагается, что запросы поступают от источника совокупности настолько большой, что она может считаться бесконечной. Результат этого допущения заключается в том, что скорость поступления запросов в систему не изменяется. Если совокупность конечна, тогда ее размер уменьшается по мере того, как запросы поступают в систему. При этом, как правило, скорость поступления запросов пропорционально уменьшается. Обычно сетевые и серверные проблемы могут решаться при допущении о бесконечной совокупности.

♦ Размер очереди (queue size). Предполагается бесконечный размер очереди, то есть очередь может расти неограниченно. Если очередь конечна, некоторые запросы могут теряться системой — когда очередь заполнена и поступает дополнительный запрос, система вынуждена отбросить один из запросов. На практике всякая очередь конечна, но во многих случаях это не оказывает существенного влияния на анализ. Мы кратко рассмотрим вопрос конечности очереди далее в этой главе.

♦ Дисциплина диспетчеризации (dispatching discipline). Когда сервер завершает обработку очередного запроса, а в очереди содержится больше одного запроса, должно быть принято решение о механизме выбора из очереди следующего запроса. Простейший метод диспетчеризации представляет собой метод FIFO (First in First Out — первым прибыл, первым обслужен). Как правило, этот метод применяется, когда используется термин очередь. Другой возможный вариант — метод LIFO (Last In First Out — последним прибыл, первым обслужен). Общий подход заключается в использовании дисциплины диспетчеризации, основанной на относительных приоритетах. Например, маршрутизатор может учитывать информацию о качестве обслуживания.

Ниже перечислены некоторые полезные параметры, включая показанные на рис. 8.2, а также некоторые другие (интерес, в частности, часто представляет вариация различных параметров, которая выражается среднеквадратичным отклонением):

♦ λ — скорость поступления, то есть среднее количество поступающих в секунду запросов;

♦ T_s — среднее время обслуживания каждого запроса; в это время не входит время ожидания в очереди;

- ♦ σ_{T_s} — среднее квадратичное отклонение времени обслуживания;
- ♦ σ — коэффициент использования; доля времени, которую сервер (серверы) занят;
- ♦ u — интенсивность трафика;
- ♦ r — среднее количество запросов в системе, ожидающих и обслуживаемых;
- ♦ R — количество запросов в системе, ожидающих и обслуживаемых;
- ♦ T_r — среднее время, которое запрос проводит в системе;
- ♦ T_R — время, которое запрос проводит в системе;
- ♦ σ_r — среднее квадратичное отклонение r ;
- ♦ σ_{T_r} — среднее квадратичное отклонение T_r ;
- ♦ w — среднее количество запросов, ожидающих обслуживания;
- ♦ σ_w — среднее квадратичное отклонение w ;
- ♦ T_w — среднее время ожидания (включая запросы с нулевым временем ожидания);
- ♦ T_d — среднее время ожидания (исключая запросы с нулевым временем ожидания);
- ♦ N — количество серверов;
- ♦ $m_x(y)$ — y -й процентиль; это значение, ниже которого величина x встречается с частотой y .

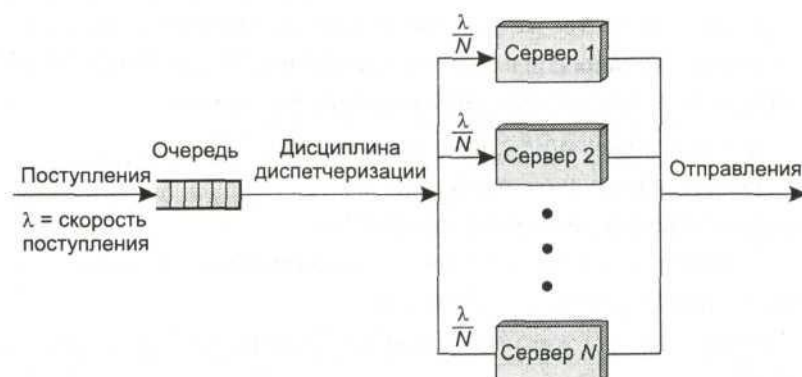
Очередь к нескольким серверам

На рисунке, а показано обобщение обсуждавшейся ранее простой модели для нескольких серверов с общей очередью. Если прибывает запрос и свободен хотя бы один сервер, тогда этот запрос немедленно передается этому серверу. Предполагается, что все серверы идентичны. Таким образом, если свободными одновременно оказываются несколько серверов, то выбор конкретного сервера не влияет на время обслуживания. Если все серверы заняты, начинает формироваться очередь. Как только один из серверов освобождается, из очереди с использованием дисциплины диспетчеризации выбирается запрос и передается серверу.

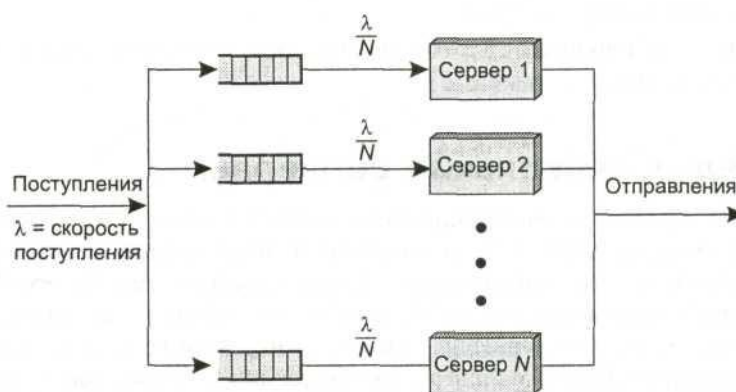
Все параметры, кроме коэффициента использования, показанные на рисунке, с тем же успехом применимы к случаю нескольких серверов. Если у нас есть N идентичных серверов, тогда ρ представляет собой коэффициент использования каждого сервера, а $N\rho$ можно рассматривать как коэффициент использования системы в целом. Этот термин часто называют интенсивностью трафика (intensity traffic) и. Таким образом, теоретический максимум коэффициента использования составляет $N \cdot 100\%$, а теоретический максимум для входной скорости равен

$$\lambda_{\max} = \frac{N}{T_s}.$$

Как правило, ключевые характеристики, выбираемые для очереди к нескольким серверам, соответствуют характеристикам очереди к одному серверу. Это означает, что мы предполагаем бесконечную совокупность и бесконечный размер общей для всех серверов очереди. За исключением специально оговариваемых случаев, дисциплина диспетчеризации представляет собой порядок FIFO.



а



б

Сравнение моделей одной очереди: а — к нескольким серверам, б — к каждому серверу

На рисунке б, напротив, представлена структура из отдельной очереди к каждому из нескольких серверов. Как будет показано ниже, это незначительное изменение структуры оказывает существенное влияние на производительность.

Основные соотношения теории очередей

Чтобы продвинуться дальше, мы должны принять несколько упрощающих допущений. Принимая эти допущения, мы рискуем сделать модель менее соответствующей реальной ситуации. К счастью, во многих случаях результаты остаются достаточно точными для задач планирования и проектирования.

Однако существуют соотношения, справедливые для общего случая. Эти соотношения приведены в табл. 2.4. Сами по себе они не очень полезны, однако могут использоваться для ответов на несколько основных вопросов. Вот пример, взятый из [154]. Рассмотрим шпиона из закусочной «Королевские Гамбургеры», пытающегося вычислить количество людей в закусочной McDonald's, расположенной на противоположной стороне улицы. Он не может весь день сидеть в McDonald's, поэтому должен найти ответ, основываясь только на наблюдении входящих в McDonald's и выходящих оттуда людей. Пронаблюдав за дверью конкурирующей фирмы в течение дня, он определяет, что за час в среднем эту закусочную посещают 32

посетителя. Он также замечает, что средний посетитель остается внутри 12 минут. С помощью формулы Литтла (Little) шпион определяет, что в каждый момент времени в McDonald's находится в среднем 6,4 посетителя ($6,4 = 32 \text{ посетителя в час} \cdot 0,2 \text{ часа/посетителя}$).

Таблица 2.4. Некоторые основные соотношения теории очередей

Общие	Один сервер	Несколько серверов
$r = \lambda T_r$ — формула Литтла	$\rho = \lambda T_s$	$\rho = \frac{\lambda T_s}{N}$
$r = \lambda T_r$ — формула Литтла	$r = w + \rho$	$u = \lambda T_s = \rho N$
$T_r = T_w + T_s$		$r = w + N\rho$

Чтобы понять формулы Литтла, рассмотрим следующий пример с одним запросом. Когда запрос поступает в систему, он обнаруживает, что в среднем w других запросов ждут в очереди впереди него. Когда запрос покидает очередь, отправляясь на обработку, за ним в очереди остается в среднем все те же w ожидающих запросов. Чтобы нагляднее представить это, обратите внимание на то, что пока запрос ждет своей очереди, очередь перед ним уменьшается, а позади него растет. Когда запрос покидает очередь, среднее количество запросов позади него равно w , так как w , по определению, равно среднему количеству запросов, ожидающих обслуживания. Среднее время ожидания обслуживания равно T_w . Поскольку запросы прибывают со скоростью λ , мы можем считать, что за время T_w должно поступить λT_w запросов. Таким образом, $w = \lambda T_w$. Путем аналогичных рассуждений можно вывести формулу $r = \lambda T_r$,

Теперь рассмотрим последнюю формулу в первом столбце таблицы. Несложно видеть, что время, которое запрос проводит в системе, представляет собой сумму времени ожидания и времени обслуживания. Таким образом, в среднем, $T_r = T_w + T_s$. Последние формулы во втором и третьем столбцах можно легко проверить. В любой момент времени число запросов в системе представляет собой сумму количеств ожидающих в очереди и обслуживаемых запросов. Для одного сервера среднее число обслуживаемых запросов равно ρ , поэтому $r = w + \rho$. Аналогично, для N серверов $r = w + N\rho$.

Допущения

Входной информацией для очереди является:

- ◆ скорость поступления запросов;
- ◆ время обслуживания; ◆ количество серверов.

Фундаментальная задача анализа очередей заключается в том, чтобы на основании такой входной информации получить следующую информацию на выходе:

- ◆ количество ожидающих запросов;
- ◆ время ожидания;

- ♦ количество запросов в очереди;
- ♦ время нахождения запроса в системе.

Что конкретно нам нужно знать об этих величинах? Во-первых, нам нужно знать их средние значения (w , T_w , r , T_r). Кроме того, было бы полезно кое-что узнать об их изменчивости. То есть нам нужно знать среднеквадратичное отклонение каждой из этих величин ($\sigma_r, \sigma T_r, \sigma r, \sigma T_w$). Также могут быть полезны и другие величины. Например, при проектировании буфера маршрутизатора или мультиплексора, возможно, будет полезно знать размер буфера, вероятность переполнения которого не превосходит 0,001. Другими словами, чему равно N , такое, что $P\{ \text{число ожидающих запросов} < N \} = 0,999$?

Чтобы отвечать на подобные вопросы, в общем случае требуется полное знание распределения вероятности временных интервалов между поступлениями запросов (интервалов времени между успешными поступлениями запросов) и времени обслуживания. Более того, даже при наличии этой информации получающиеся в результате формулы оказываются исключительно сложными. Поэтому, чтобы двигаться дальше, мы должны принять некоторые упрощающие допущения.

Наиболее важное из этих допущений касается скорости поступления запросов. Предполагается, что интервалы времени между поступлениями запросов распределены экспоненциально. Это равносильно заявлению о подчинении числа поступающих за период времени t запросов распределению Пуассона. Другими словами, это означает, что запросы поступают случайным образом и не зависят друг от друга. Это допущение принимается почти всегда. Без него анализ очередей оказывается невозможным или, в крайнем случае, довольно сложным. При таком допущении оказывается, что, зная только среднее значение и среднеквадратичное отклонение скорости поступления запросов, можно получить много полезных результатов. Ситуацию можно упростить еще больше, если предположить, что время обслуживания подчиняется экспоненциальному распределению или постоянно. При этом могут быть получены еще более точные результаты.

Для обозначения основных допущений, принимаемых при моделировании очередей, была разработана так называемая нотация Кендалла (Kendall's notation). Эта нотация имеет вид $X/Y/N$, где X обозначает распределение интервалов времени между поступлениями запросов, Y — распределение времени обслуживания, N — количество серверов. Наиболее часто встречающиеся распределения обозначаются следующим образом:

G — произвольное распределение интервалов времени обслуживания запросов или интервалов времени между поступлениями запросов;

GI — произвольное распределение интервалов времени между поступлениями запросов с ограничением, заключающимся в независимости этих интервалов;

M — отрицательное экспоненциальное распределение;

D — детерминированное поступление запросов, или обслуживание фиксированной длины.

Таким образом, запись M/M/1 означает модель с одним сервером, в которой количество поступающих запросов распределено по Пуассону (интервалы времени между поступлениями распределены экспоненциально), и с экспоненциальным распределением интервалов времени обслуживания.

Очередь к одному серверу

Ниже приведены некоторые формулы для очереди к одному серверу при следующих допущениях:

частота поступления запросов подчиняется распределению Пуассона;
дисциплина диспетчеризации не дает предпочтения запросам,

основываясь

на времени обслуживания;

в формулах для среднеквадратичного отклонения предполагается диспетчеризация FIFO;

запросы не выбрасываются из очереди.

Для первого случая частота поступления запросов распределена по Пуассону, а интервалы времени обслуживания подчиняются произвольному распределению. Использование коэффициента масштабирования A позволяет упростить формулы для некоторых ключевых выходных переменных. Обратите внимание на то, что масштабирующий коэффициент зависит от отношения среднеквадратичного отклонения времени обслуживания к среднему значению. Никакой другой информации о времени обслуживания не требуется. Итак, формулы для произвольного распределения времени обслуживания (M/G/1):

$$A = \frac{1}{2} \left[1 + \left(\frac{\sigma_r}{T_s} \right)^2 \right];$$

$$r = \rho + \frac{\rho^2 A}{1 - \rho};$$

$$w = \frac{\rho^2 A}{1 - \rho};$$

$$T_r = T_s + \frac{\rho T_s A}{1 - \rho};$$

$$T_w = \frac{\rho T_s A}{1 - \rho}.$$

Интерес представляют два особых случая. Когда среднеквадратичное отклонение равно среднему значению, распределение времени обслуживания является экспоненциальным (M/M/1). Это простейший случай, в котором легче всего получить результаты. Далее показаны упрощенные варианты формул для среднеквадратичного отклонения r и T_r а также некоторые другие интересные соотношения:

$$r = \frac{\rho}{1-\rho}; w = \frac{\rho^2}{1-\rho};$$

$$T_r = \frac{T_s}{1-\rho}; T_w = \frac{\rho T_s}{1-\rho};$$

$$\sigma_r = \frac{\sqrt{\rho}}{1-\rho}; \sigma_{T_r} = \frac{T_s}{1-\rho};$$

$$\Pr[R = N] = (1-\rho)\rho^N;$$

$$\Pr[R \leq N] = \sum_{i=0}^N (1-\rho)\rho^i;$$

$$\Pr[T_R \leq T] = 1 - e^{-(1-\rho)t/T_s},$$

$$m_{T_r}(y) = T_r \cdot \ln\left(\frac{100}{100-y}\right);$$

$$m_{T_w}(y) = \frac{T_w}{\rho} \cdot \ln\left(\frac{100\rho}{100-y}\right).$$

Другой интересный случай, когда среднее квадратичное отклонение времени обслуживания равно нулю, что означает постоянное время обслуживания (M/D/1):

$$r = \frac{\rho^2}{2(1-\rho)} + \rho;$$

$$w = \frac{\rho^2}{2(1-\rho)};$$

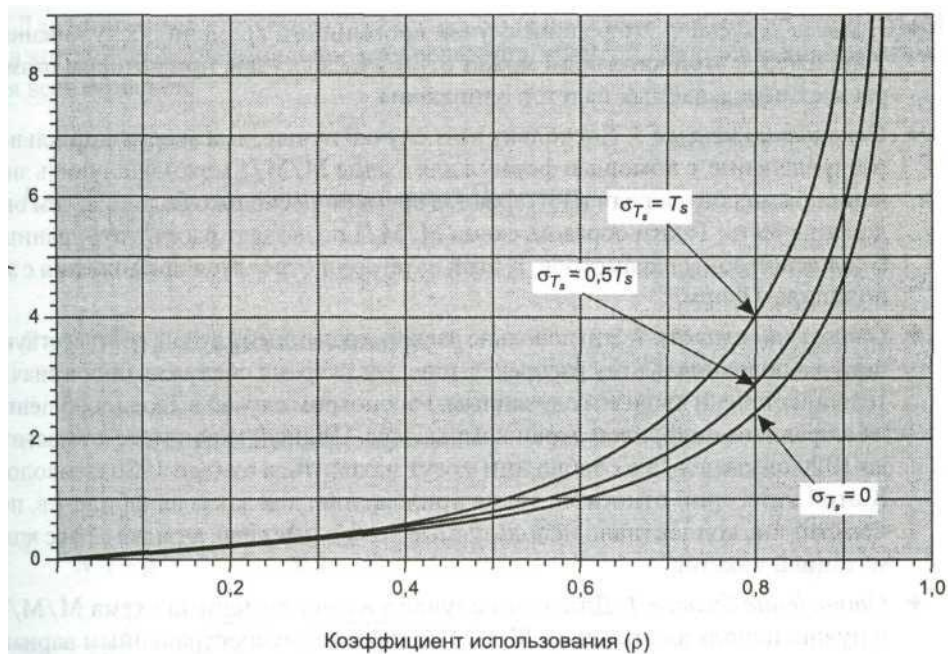
$$T_r = \frac{T_s(2-\rho)}{2(1-\rho)};$$

$$T_w = \frac{\rho T_s}{2(1-\rho)};$$

$$\sigma_r = \frac{1}{1-\rho} \sqrt{\rho - \frac{3\rho^2}{2} + \frac{5\rho^3}{6} - \frac{\rho^4}{12}};$$

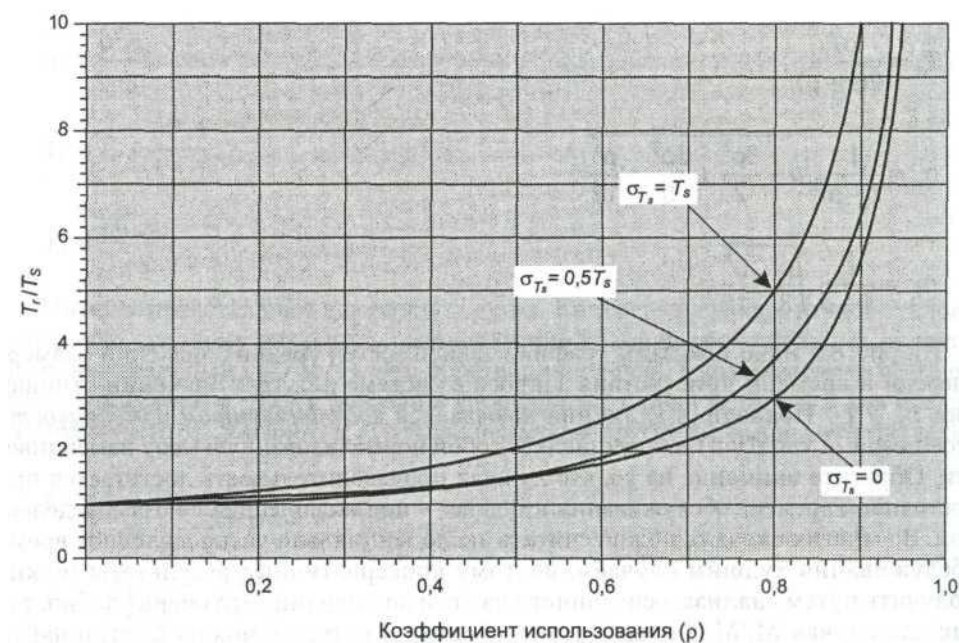
$$\sigma_{T_r} = \frac{T_s}{1-\rho} \sqrt{\frac{\rho}{3} - \frac{\rho^2}{12}}.$$

На рисунке показаны графики зависимостей средних значений размера очереди и времени присутствия запроса в системе для трех значений отношения $\sigma T_s / T_s$. Последняя величина называется коэффициентом изменчивости (coefficient of variation) и представляет собой нормализованную меру изменчивости. Обратите внимание на то, что лучшая производительность достигается при постоянном времени обслуживания, а худшая — при экспоненциально распределенном. Во многих случаях можно считать экспоненциально распределенное время обслуживания худшим случаем, поэтому консервативные результаты можно получить путем анализа, основанного на этом допущении. Это очень удобно, так как для случая M/M/1 существуют таблицы, в которых можно быстро найти нужные значения.



Среднее число запросов в системе для очереди к одному серверу

Какое отношение aTs / Ts может встретиться на практике? Мы можем рассматривать четыре области значений:



Среднее резидентное время для очереди к одному серверу

Нулевое значение. Это редкий случай постоянного времени обслуживания. Например, к этой категории может относиться случай, при котором размеры всех передаваемых пакетов одинаковы.

Отношение меньше 1. Поскольку этот случай лучше, чем экспоненциальное распределение, с помощью формул для схемы М/М/1 можно получить значения размеров очереди и интервалов времени несколько большие, чем они должны быть. Таким образом, схема М/М/1 позволяет рассчитать границы безопасной зоны параметров. К этой категории относятся приложения с заполнением форм.

Отношение близко к 1. Это довольно распространенный случай, соответствующий экспоненциальному распределению, когда время обслуживания в значительной степени является случайным. Рассмотрим случай вывода сообщений на алфавитно-цифровой экран компьютера. Полный экран может вместить до 1920 символов, а в сообщении могут находиться от 0 до 1920 символов. К этой категории относятся такие приложения, как заказ авиабилетов, поиск файлов, коллективно используемые локальные сети, а также сети с коммутацией пакетов.

Отношение больше 1. Для этого случая уже неприменима схема $M/M/1$, и нужно использовать схему $M/G/1$. Наиболее распространенным вариантом данного случая является бимодальное распределение с далеко отстоящими друг от друга пиками. Пример такого случая — система с большим количеством коротких сообщений, большим количеством длинных сообщений и практически без сообщений промежуточного размера.

Те же соображения применимы к скорости поступления запросов. Если частота поступления запросов распределена по Пуассону, то интервалы времени подчиняются экспоненциальному распределению, а отношение среднеквадратичного отклонения к среднему значению равно 1. Если данное отношение существенно меньше 1, тогда поступления стремятся к равноотстоящему варианту (с невысоким уровнем изменчивости), и приближение Пуассона позволяет получить оценку верхних границ размеров очереди и времени ожидания. С другой стороны, если это отношение больше 1, тогда запросы поступают очень неравномерно и проблема перегрузки становится очень актуальной.

Очередь к нескольким серверам

Ниже перечислены формулы для некоторых ключевых параметров случая нескольких серверов. Ряд ограничений накладывается в допущениях: частота поступления запросов подчиняется распределению Пуассона; значения времени обслуживания распределены экспоненциально; все серверы загружены в равной мере; среднее время обслуживания всех серверов одинаково; используется диспетчеризация FIFO; запросы не выбрасываются из очереди.

Для этой модели полезная статистика была получена только для схемы М/М/ N, в которой экспоненциальное распределение времени обслуживания идентично для всех серверов:

$$K = \frac{\sum_{l=0}^{N-1} \frac{(N\rho)^l}{l!}}{\sum_{l=0}^{\infty} \frac{(N\rho)^l}{l!}} - \text{функция отношения Пуассона};$$

$$C = \frac{1-K}{1-\rho K} - \text{С-функция Эрланга};$$

$$r = C \frac{\rho}{1-\rho} + N\rho; \quad w = C \frac{\rho}{1-\rho};$$

$$T_r = \frac{C}{N} \frac{T_s}{1-\rho} + T_s; \quad T_w = \frac{C}{N} \frac{T_s}{1-\rho};$$

$$\sigma_{T_s} = \frac{T_s}{N(1-\rho)} \sqrt{C(2-C) + N^2(1-\rho)^2};$$

$$\sigma_{T_s} = \frac{1}{1-\rho} \sqrt{C\rho(1+\rho) - C\rho};$$

$$\Pr[T_w > t] = Ce^{-N(1-\rho)t/T_s};$$

$$m_{T_s}(r) = \frac{T_s}{N(1-\rho)} \ln\left(\frac{100C}{100-r}\right);$$

$$T_d = \frac{T_s}{N(1-\rho)}.$$

Обратите внимание на присутствие С-функции Эрланга практически во всех уравнениях. Она означает вероятность того, что в данный момент времени заняты все серверы. Другими словами, это вероятность того, что количество запросов в системе (ожидających и обслуживаемых) больше или равно количеству серверов. Соответствующее уравнение имеет вид

$$C(N, \rho) = \frac{1 - K(N, \rho)}{1 - \rho K(N, \rho)}.$$

Здесь K — функция отношения Пуассона. Поскольку величина C представляет собой вероятность, эта величина всегда находится в диапазоне от 0 до 1. Как можно видеть, она является функцией от числа серверов и коэффициента использования. Это выражение часто встречается при расчетах очередей. Можно воспользоваться готовыми табличными

значениями или компьютерной программой. Обратите внимание на то, что для системы с одним сервером эта формула упрощается до $C(1, p) = p$.

Глава 3

Описание структурной схемы контроллера

3.1 Основные элементы

Центральный процессор K1810BM86

NMI INTR	CPU	AD 0
CLK		AD 1
RESET		AD 2
READY		AD 3
		AD 4
$\overline{\text{TEST}}$		AD 5
		AD 6
$\overline{\text{MN/MX}}$		AD 7
		AD 8
HLDA (RQ/GT1)		AD 9
		AD 10
HOLD (RQ/GT1)		AD 11
		AD 12
		AD 13
		AD 14
		AD 15
GND		A16/ST3
GND		A17/ST4
Ucc		A18/ST5
		A19/ST6
		$\overline{\text{BHE/ST7}}$
		$\overline{\text{RD}}$
		$\overline{\text{WR}}(\text{LOCK})$
		M/ $\overline{\text{IO}}(\text{ST2})$
		DT/ $\overline{\text{R}}(\text{ST1})$
		$\overline{\text{DEN}}(\text{ST0})$
		$\overline{\text{ALE}}(\text{QS0})$
		$\overline{\text{INTA}}(\text{QS1})$

Микропроцессор серии K1810 является дальнейшим развитием процессора KP580. Процессор K1810BM86 является 16 разрядным.

Процессор программно совместим с процессорами предыдущей серии и аппаратно совместим со всеми БИС серии KP580

Максимальный режим ориентирован на работу K1810BM86 в составе мультимикропроцессорных систем, в которых, помимо нескольких центральных процессоров K1810BM86.

Определим более четко введенные выше понятия:

Центральный процессор – поддерживает собственный командный цикл, выполняет программу, хранящуюся в системной памяти, по сбросу системы управление, как правило, передается центральному процессору (или одному из ЦП, если их несколько в системе).

Специализированный процессор – поддерживает собственный командный цикл, выполняет программу, хранящуюся в системной памяти, но инициализируется только по команде ЦП, по окончании выполнения программы сообщает ЦП о завершении работы.

Сопроцессор не поддерживает собственный командный цикл, выполняет команды, выбираемые для него ЦП из общего потока команд. По сути дела сопроцессор является расширением ЦП.

Внутренняя структура

Структурная схема МП K1810BM86 представлена на рис . МП включает в себя три основных устройства :

УОД - устройство обработки данных;

УСМ - устройство связи с магистралью;

УУС - устройство управления и синхронизации.

УОД предназначено для выполнения команд и включает в себя 16-разрядное АЛУ, системные регистры и другие вспомогательные схемы; блок регистров (РОН, базовые и индексные) и блок микропрограммного управления.

УСМ обеспечивает формирование 20-разрядного физического адреса памяти и 16-разрядного адреса ВУ, выбор команд из памяти, обмен данными с ЗУ, ВУ, другими процессорами по магистрали. УСМ включает в себя сумматор адреса, блок регистров очереди команд и блок сегментных регистров.

УУС обеспечивает синхронизацию работы устройств МП, выработку управляющих сигналов и сигналов состояния для обмена с другими устройствами, анализ и соответствующую реакцию на сигналы других устройств МПС.

МП может работать в одном из двух режимов - "минимальном" (min) и "максимальном" (max). Минимальный режим предназначен для реализации однопроцессорной конфигурации МПС с организацией, подобной МПС на базе i8080, но с увеличенным адресным пространством, более высоким быстродействием и значительно расширенной системой команд. Максимальная конфигурация предполагает наличие в системе нескольких МП и специального блока арбитра магистрали (используется интерфейс Multibus).

На внешних выводах МП K1810BM86 широко используется принцип мультиплексирования сигналов - передача разных сигналов по общим линиям с разделением во времени. Кроме того, одни и те же выводы могут использоваться для передачи разных сигналов в зависимости от режима (min - max).

Ниже приводится описание внешних выводов МП K1810BM86. При описании выводов косой чертой разделены сигналы, появляющиеся на

выводе в разные моменты машинного цикла. В круглых скобках указаны сигналы, характерные только для максимального режима. Символ \ после имени сигнала - знак его инверсии.

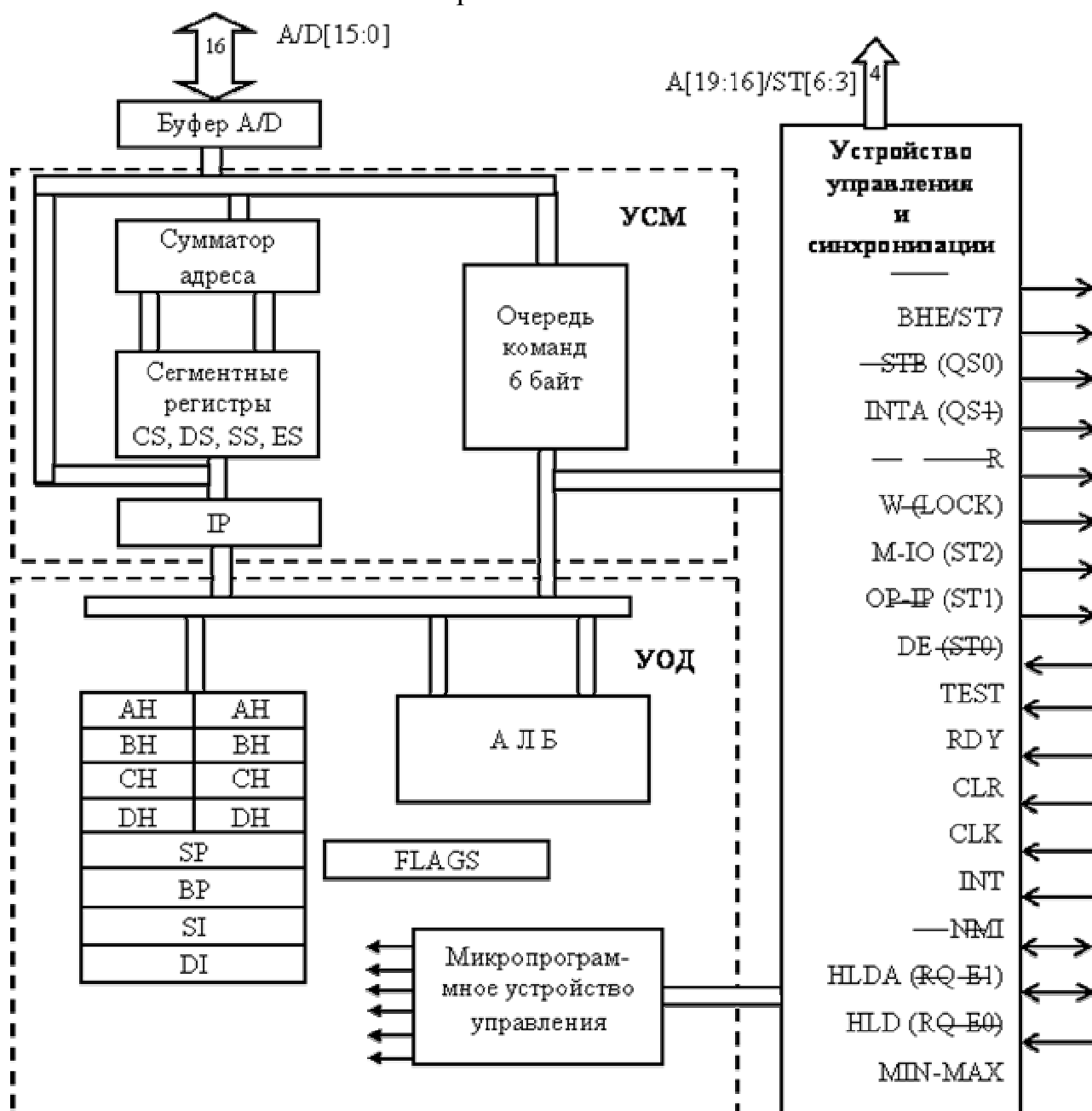


Рис.3.1. Внутренняя структура процессора K1810BM86

A/D[15:0] - младшие [15:0] разряды адреса / данные;

A[19:16]/ST[6:3] - старшие [19:16] разряды адреса / сигналы состояния;

BHE\ST[7] - разрешение передачи старшего байта данных / сигнал состояния;

STB(QS0) - строб адреса (состояние очереди команд);

R\ - чтение;

W\-(LOCK\) - запись (блокировка канала);

M-IO\-(ST2\) - память - внешнее устройство (состояние цикла);

OP-IP\-(ST1\) - выдача-прием (состояние цикла);

DE\ (ST0\) - разрешение передачи данных (сост. цикла);
 TEST\ - проверка;
 RDY - готовность;
 CLR - сброс;
 CLC - тактовый сигнал;
 INT - запрос внешнего прерывания;
 INTA\ (QS1) - подтверждение прерывания (состояние очереди команд);
 NMI - запрос немаскируемого прерывания;
 HLD(RQ\VE0) - запрос ПДП (запрос / подтверждение доступа к магистрали);
 NLDA(RQ\VE1) - подтверждение ПДП (запрос / подтверждение доступа к магистрали);
 MIN/MAX\ - потенциал задания режима (1-min, 0-max).

Сигналы состояния ("статуса") используются для отображения внутреннего состояния МП. Некоторые группы статусных сигналов используются только в максимальном режиме.

Сигналы ST[2:0]\ определяют тип текущего машинного цикла (аналогично PSW для i8080), и формируются только в максимальном режиме:

Табл.3.1

ST2	ST1	ST0	Тип машинного цикла
0	0	0	Обслуживание прерывания
0	0	1	Чтение ВУ
0	1	0	Запись ВУ
0	1	1	Останов
1	0	0	Извлечение кода команды
1	0	1	Чтение ЗУ
1	1	0	Запись ЗУ
1	1	1	Пассивное состояние

Разряды статуса 3..4 определяют сегментный регистр, используемый для вычисления физического адреса:

Табл.3.2

ST4	ST3	Сегментный регистр
0	0	ES
0	1	CS
1	0	SS
1	1	DS

ST[5] отражает состояние флага разрешения прерывания IF, ST[6] всегда установлен в 0, когда МП обменивается информацией по магистрали, состояние ST[7] не определено (зарезервировано).

Сигналы QS[1:0] формируются только в максимальном режиме и отражают состояние очереди команд:

Табл.9.3

QS1	QS0	Состояние очереди команд
0	0	Нет операции
0	1	Очередь очищается
1	0	Извлекается первый байт
1	1	Извлекается очередной байт

Практически все команды МП K1810BM86 могут работать как со словами (2 байта) так и с байтами. При работе со словами сигнал ВНЕ\ разрешает передачу старшего байта слова.

Сигнал STB отмечает наличие на линиях A/D и A/S адреса.

R и W\ стробируют данные на шине A/D соответственно при чтении и записи.

M-I/O\ и OP-IP\ определяют соответственно устройство, с которым производится обмен (память - ВУ) и направление передачи информации (вывод - ввод) относительно процессора.

DE\ стробирует внешний буфер A/D при передаче данных.

Вход TEST\ предназначен для синхронизации программы с внешними процессами. Команда WAIT (ожидание) переводит процессор в режим ожидания, в котором он будет находиться до тех пор, пока на входе TEST\ удерживается высокий уровень сигнала (лог. "1"). При этом все магистрали МП переводятся в высокоимпедансное состояние.

RDY - (готовность) аналогично соответствующему входу МП i8080 обеспечивает возможность асинхронного машинного цикла.

CLR - (сброс) устанавливает все регистры МП в 0, кроме CS, который устанавливается в FFFF и осуществляет запуск командного цикла. Таким образом, стартовый адрес K1810BM86 - всегда FFFF0.

CLC - тактовый сигнал.

INT, INTA\ - соответственно запрос и подтверждение внешнего прерывания (подробнее см. раздел 7).

NMI - запрос внешнего немаскируемого прерывания по фиксированному вектору 2.

HLD, HLDA - соответственно требование и предоставление прямого доступа в память (работает аналогично подсистеме ПДП i8080). В максимальном режиме вместо сигналов HLD, HLDA используются две двунаправленные линии RQ/Ei\ - запрос шины/разрешение доступа, на которые работает специальная микросхема - арбитр шины.

Машинные циклы K1810BM86 в минимальном и максимальном режимах

Временные диаграммы работы МП K1810BM86 в минимальном режиме представлены на Рис 3.2. и Рис. 3.3

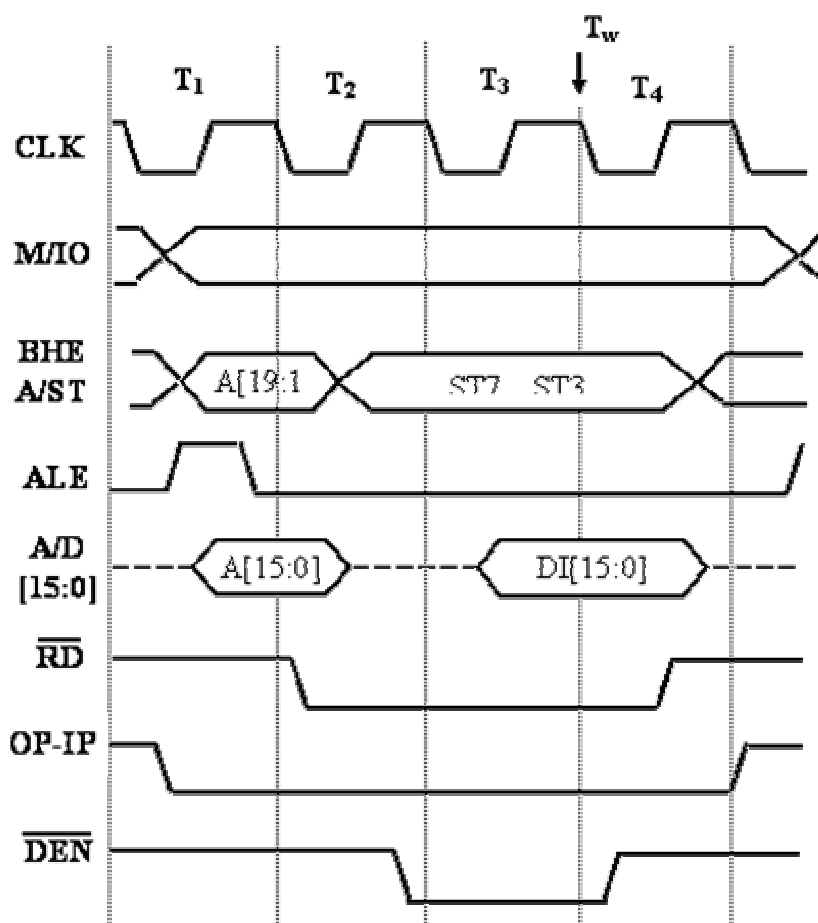


Рис.3.2. Минимальный режим – цикл ЧТЕНИЕ

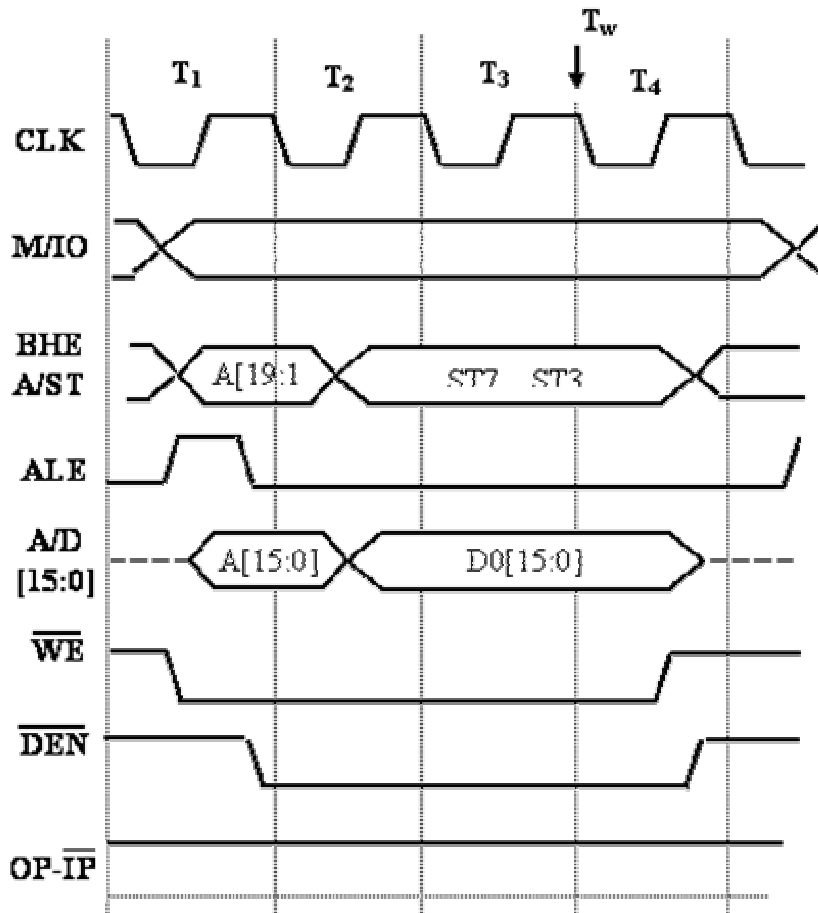


Рис.9.6. Минимальный режим – цикл ЗАПИСЬ

Цикл начинается с формирования в T1 сигнала М/Ю, определяющего тип устройства - память или ВУ, с которым осуществляется обмен данными. Длительность сигнала М/Ю равна длительности цикла шины, и он используется для селекции адреса устройств. В T1 и в начале T2 МП выдает адрес A[19:16] и A[15:0] и сигнал ВНЕ, который вместе с A0 определяет передачу слова или одного из байтов. По спаду строба ALE адрес фиксируется во внешних регистрах-защелках.

Шинные формирователи КР580ВА86

A0	BD	B0
A1		B1
A2		B2
A3		B3
A4		B4
A5		B5
A6		B6
A7		B7
T		Ucc
OE		GND

Микросхема КР580ВА86 представляет собой двунаправленный 8-разрядный шинный формирователь с высокой нагрузочной способностью, позволяет осуществить связь микропроцессора с периферийными устройствами ввода/вывода информации.

Режим работы микросхемы КР580ВА86 определяется управляющими сигналами OE «Разрешение входа» и Т «Направление передачи». При поступлении на вход OE сигнала высокого уровня информационные выходы А и В переходят в состояние «Выключено». При наличии на входе OE сигнала низкого уровня направление передачи информации определяется сигналом Т. При подаче на вход Т сигнала высокого уровня осуществляется передача информации с канала А в В, при подаче на вход Т сигнала низкого уровня наоборот, с канала В в канал А.

Назначение выводов:

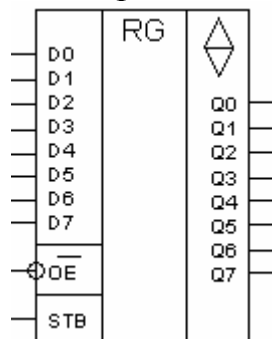
АО-А7 - Шина А (вход/выход);

OE - Разрешение выхода;

Т - Направление передачи;

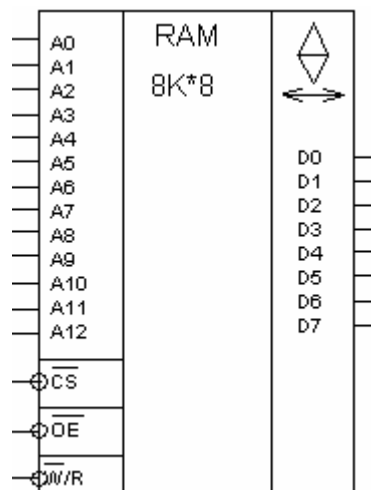
ВО-В7 - Шина В (вход/выход).

Регистр КР580ИР82



Микросхема КР580ИР82 8-разрядный адресный регистр, предназначен для связи микропроцессора с системной шиной; обладает повышенной нагрузочной способностью. Микросхема КР580ИР82 — 8-разрядный О-регистр-«защелка» без инверсии и с тремя состояниями на выходе.

Статическое ОЗУ K537PY12



Для организации ОЗУ МПС будем использовать одну микросхему RAM типа K537PY12, имеющую организацию 8К*8.

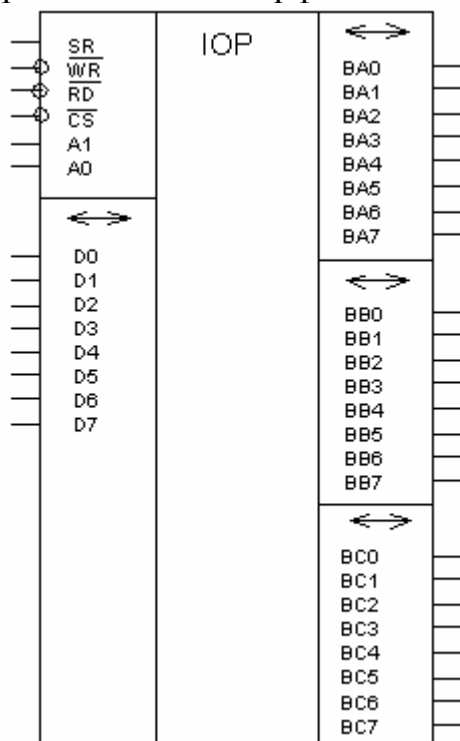
Назначение управляющих входов:

CS – выбор кристалла;

OE разрешение чтения;

WR - разрешение записи.

Программируемый параллельный интерфейс KP580BB55A



Микросхема KP580BB55A предназначена для параллельной передачи информации между микропроцессором и периферийными устройствами и содержит три 8-разрядных канала ввода/вывода А, В, С.

Канал С может быть представлен в виде двух 4-разрядных каналов ввода/вывода, доступ к которым производится как к отдельным независимым каналам, периферийные устройства подключаются к каналам А, В, С, связь с микропроцессором осуществляется с помощью шины D через буфер данных. Каждый из каналов А, В, С состоит из 8-разрядного регистра и двунаправленных формирователей, имеющих на выходе состояние «Выключено». Устройство управления содержит регистр управляющего слова (РУС), в который предварительно производится запись информации, определяющей режим работы каналов, и формирует сигналы выбора канала и управления каналом С.

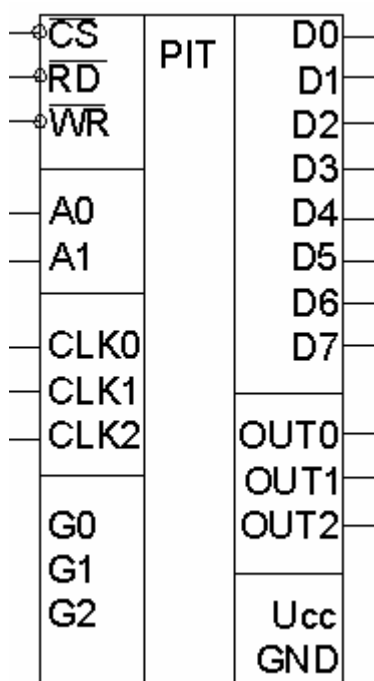
Микросхема может работать в одном из трех режимов: режим 0 — простой ввод/вывод; режим 1 — стробируемый ввод/вывод; режим 2 — двунаправленный канал. Режим работы каналов можно изменять как в начале, так и в процессе выполнения программы. Каналы А и В могут работать в различных режимах, а работа канала С зависит от режимов работы каналов А и В. Комбинируя режимы работы каналов, можно обеспечить работу микросхемы почти с любым периферийным устройством.

В режиме 0 осуществляется простой ввод/вывод данных по трем 8-разрядным каналам, причем канал С может использоваться как два 4-разрядных канала. Каждый из каналов может использоваться отдельно для ввода или вывода информации. В режиме 0 входная информация не запоминается, а выходная хранится в выходных регистрах до записи новой информации в канал или до записи нового режима.

В режиме 1 передача данных осуществляется только через каналы А и В, а линии канала С служат для приема и выдачи сигналов управления. Каждый из каналов А и В независимо друг от друга может использоваться для ввода или вывода 8-разрядных данных, причем входные и выходные данные фиксируются в регистрах каналов.

В режиме 2 для канала А обеспечивается возможность обмена информацией с периферийными устройствами по 8-разрядному двунаправленному каналу. Для организации обмена используется пять линий канала С. В режиме 2 входные и выходные данные фиксируются во входном и выходном регистрах соответственно.

Программируемый таймер К1810ВИ54



Программируемый таймер (ПТ) К1810ВИ54 предназначен для генерации временных функций, программно-управляемых временных задержек с возможностью программного контроля их выполнения.

Программируемые таймеры применяются в МПС, выполненных на базе МПК БИС К580, К1810, К1821, используемых в задачах управления и измерения в реальном времени с тактовой частотой до 8 МГц.

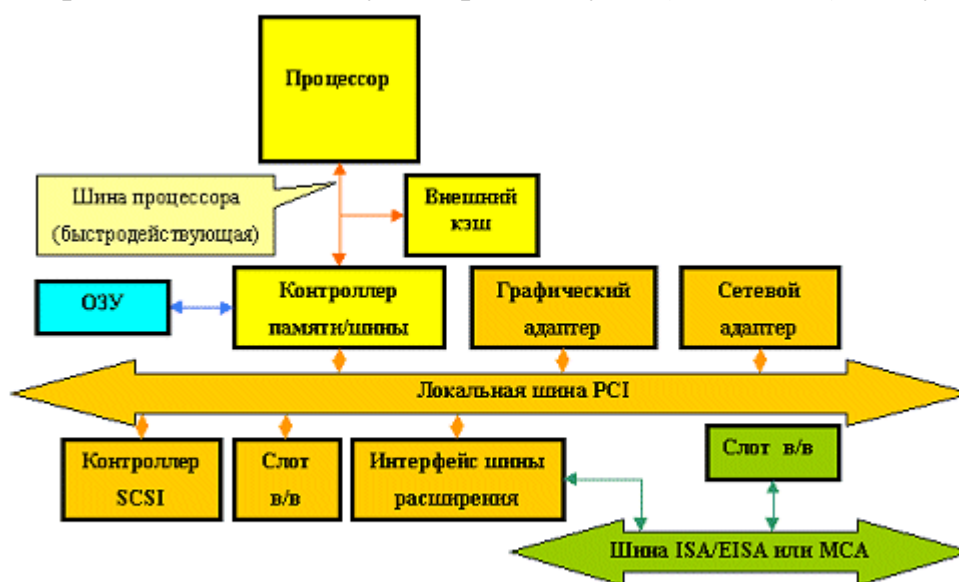
Конструктивно эти ПТ совместимы с ПТ типа К1810ВИ53, отличаются от них повышенным быстродействием и расширенными функциональными возможностями.

Программируемый таймер включает в себя 3 независимых канала, каждый из которых может быть запрограммирован на работу в одном из шести режимов для двоичного или двоично-десятичного счета.

3.2 Интерфейс PCI

Итак, переходим к самому интересному. Что же находится на сегодняшний день внутри большинства наших компьютеров? Естественно, шина PCI. Другой вопрос, почему именно эта шина. Попробуем разобраться.

Итак, разработка шины PCI началась весной 1991 года как внутренний проект корпорации Intel (Release 0.1). Специалисты компании поставили перед собой цель разработать недорогое решение, которое бы позволило полностью реализовать возможности нового поколения процессоров 486/Pentium/P6 (вот уже половина ответа). Особенно подчеркивалось, что разработка проводилась "с нуля", а не была попыткой установки новых "заплат" на существующие решения. В результате шина PCI появилась в июне 1992 года (R1.0). Разработчики Intel отказались от использования шины процессора и ввели еще одну "антресольную" (mezzanine) шину.



Архитектура компьютера с шиной PCI

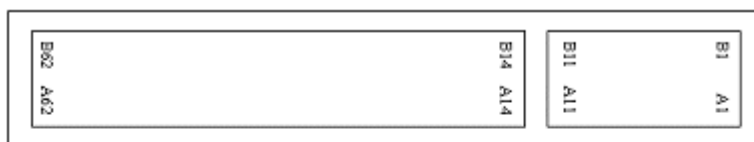
Благодаря такому решению шина получилась, во-первых, процессоронезависимой (в отличие от VLbus), а во-вторых, могла работать параллельно с шиной процессора, не обращаясь к ней за запросами. Например, процессор работает себе с кэшем или системной памятью, а в это время по сети на винчестер пишется информация. Просто здорово! На самом деле идиллии, конечно, не получается, но загрузка шины процессора снижается здорово. Кроме того, стандарт шины был объявлен открытым и передан PCI Special Interest Group, которая продолжила работу по совершенствованию шины (в настоящее время доступен R2.1), и в этом, пожалуй, вторая половина ответа на вопрос "почему PCI?"

Основные возможности шины следующие.

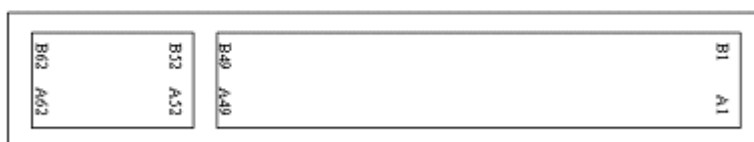
Синхронный 32-х или 64-х разрядный обмен данными (правда, насколько мне известно, 64-разрядная шина в настоящее время используется только в Alpha-системах и серверах на базе процессоров Intel Xeon, но, в принципе, за ней будущее). При этом для уменьшения числа контактов (и

стоимости) используется мультиплексирование, то есть адрес и данные передаются по одним и тем же линиям.

Поддержка 5V и 3.3V логики. Разъемы для 5 и 3.3V плат различаются расположением ключей



Разъем 32-разрядной шины PCI с напряжением питания 5V



Разъем 32-разрядной шины PCI с напряжением питания 3.3V

Существуют и универсальные платы, поддерживающие оба напряжения. Заметим, что частота 66MHz поддерживается только 3.3V логикой.

Частота работы шины 33MHz или 66MHz (в версии 2.1) позволяет обеспечить широкий диапазон пропускных способностей (с использованием пакетного режима):

- 132 MB/сек при 32-bit/33MHz;

- 264 MB/сек при 32-bit/66MHz;

- 264 MB/сек при 64-bit/33MHz;

- 528 MB/сек при 64-bit/66MHz.

При этом для работы шины на частоте 66MHz необходимо, чтобы все периферийные устройства работали на этой частоте.

Полная поддержка multiply bus master (например, несколько контроллеров жестких дисков могут одновременно работать на шине).

Поддержка write-back и write-through кэша.

Автоматическое конфигурирование карт расширения при включении питания.

Спецификация шины позволяет комбинировать до восьми функций на одной карте (например, видео + звук и т.д.).

Шина позволяет устанавливать до 4 слотов расширения, однако возможно использование моста PCI-PCI для увеличения количества карт расширения.

PCI-устройства оборудованы таймером, который используется для определения максимального промежутка времени, в течении которого устройство может занимать шину.

При разработке шины в ее архитектуру были заложены передовые технические решения, позволяющие повысить пропускную способность.

Шина поддерживает метод передачи данных, называемый "linear burst" (метод линейных пакетов). Этот метод предполагает, что пакет информации считывается (или записывается) "одним куском", то есть адрес автоматически увеличивается для следующего байта. Естественным образом при этом увеличивается скорость передачи собственно данных за счет уменьшения числа передаваемых адресов.

-INTR A,B,C,D - линии запросов прерывания, направляются на доступные линии IRQ BIOS компьютера. Запрос по низкому уровню допускает разделяемое использование линий прерывания.

Clock - сигнал синхронизации на тактовой частоте шины.

Test Clock, -TSTRES, TestDO, TestDI - сигналы для тестирования адаптеров по интерфейсу JTAG (на системной плате обычно не задействованы).

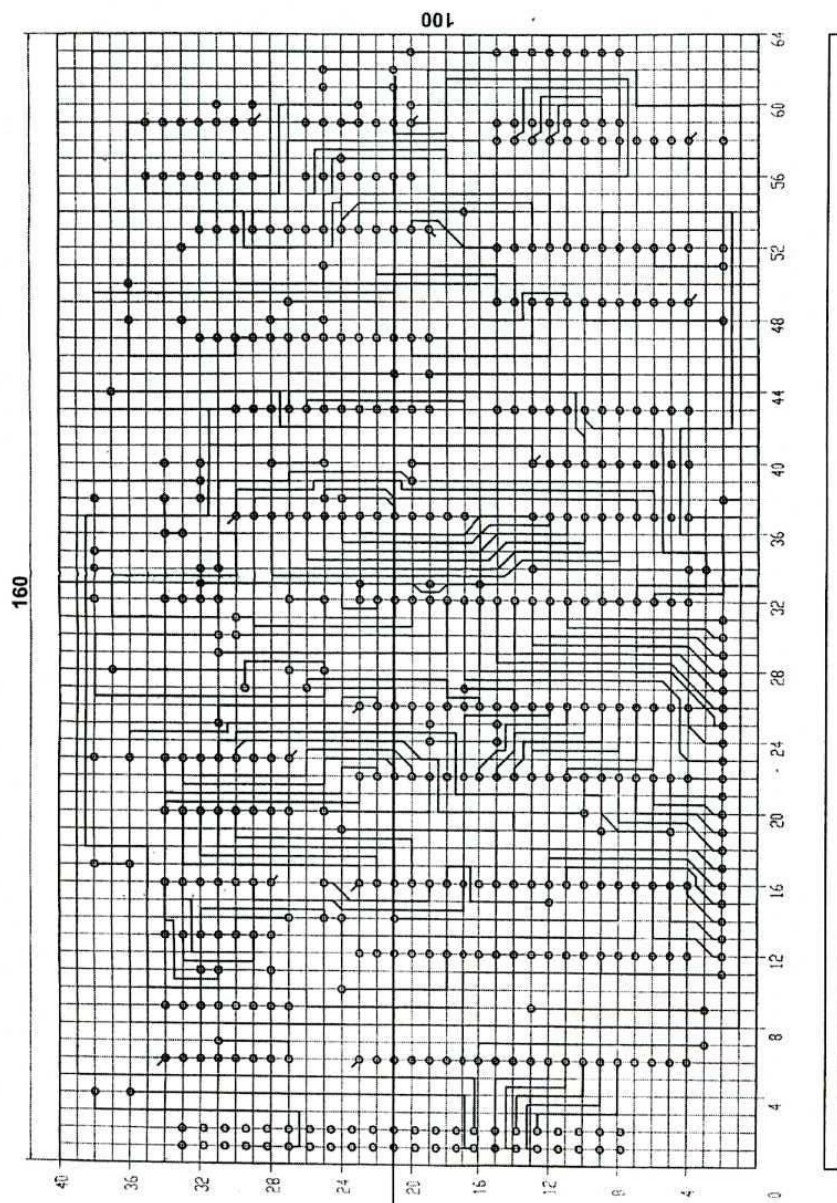
TSTMSLCT - перевод в режим тестирования.

3.3 Разработка печатной платы

Существует три основных способа проектирования и подготовки производства печатных плат. Первый способ предназначен для проектирования наиболее простых плат, для описания которых нет необходимости создания графического или текстового описания схемы принципиальной электрической. Основная подготовка выполняется в графическом редакторе PC-CARDS. Средствами редактора изображается печатная плата, на которой в ручном режиме размещаются элементы посадочных мест из технологической библиотеки файлов *.PRT. Списки цепей связности формируются непосредственным указанием на контактные выводы компонентов. Полученное описание может обрабатываться программой автоматической трассировки PC-ROUTE.

Второй способ предназначен для текстового описания цепей связности и компонентов. Этот способ целесообразно использовать для этапа подготовки производства печатной платы, когда схема электрическая принципиальная окончательно завершена и конструктор-пользователь системы PCAD, не может вносить в нее изменения. В этом случае средствами текстового редактора создается файл типа *.ALT, который обрабатывается программой PC-NLT, и полученный файл типа *.PKG может быть обработан программой PC-PLACE для размещения компонентов на печатной плате и затем программой автоматической трассировки PC-ROUTE.

Третий способ заключается в создании графического описания электрической схемы средствами графического редактора PC-CAPS. Этот способ является наиболее полным из трех анализируемых способов и может включать в этапы проектирования такие фазы, как логическое моделирование схемы электрической принципиальной, иерархическое проектирование и автоматическую упаковку логических элементов в микросхемы. В процессе проектирования выпускается схема принципиальная электрическая. Этот документ становится основой для проекта и все последующие этапы прохождения проекта, включая топологию печатной платы, могут быть автоматически сверены со схемой с диагностикой возможных ошибок. Пример разведенной платы приведен на рис



Печатная плата

Глава 4

Моделирование контроллера на ПЛИС

САПР «MAX+plus II» представляет собой интегрированную среду для разработки цифровых устройств на базе программируемых логических интегральных схем (ПЛИС) фирмы «Альтера» и обеспечивает выполнение всех этапов, необходимых для выпуска готовых изделий:

- создание проектов устройств;
- синтез структур и трассировку внутренних связей ПЛИС;
- подготовку данных для программирования или конфигурирования ПЛИС (компиляцию);
- верификацию проектов (функциональное моделирование и временной анализ);
- программирование или конфигурирование ПЛИС.

Наиболее полное фирменное описание системы «MAX+plus II» содержится на сайте фирмы «Альтера», в который входит учебник («Tutorial»). На русском языке достаточно подробное описание системы «MAX+plus II». В настоящей статье приводятся в минимальном объеме сведения, необходимые для начального освоения технологии разработки цифровых устройств на ПЛИС фирмы «Альтера». В качестве примера рассмотрен процесс разработки устройства управления 7-сегментным светодиодным индикатором.

4.1 Приложения системы «MAX+plus II»

В состав пакета «MAX+plus II» входят следующие связанные между собой приложения, реализующие все перечисленные выше этапы разработки цифровых устройств на ПЛИС фирмы «Альтера»:

Приложения для ввода проектов (редакторы проектов)

«Graphic Editor» – графический редактор, предназначенный для ввода проекта в виде схемы соединений символов элементов, извлекаемых из стандартных библиотек пакета либо из библиотеки пользователя.

«Waveform Editor» – редактор временных диаграмм (некоторые авторы называют это приложение сигнальным редактором), который выполняет двойную функцию: на этапе ввода обеспечивает ввод логики проекта в виде диаграмм (эпюр) состояний входов и выходов, а на этапе моделирования обеспечивает ввод диаграмм тестовых (эталонных) входных состояний моделируемого устройства и задание перечня тестируемых выходов.

«Text Editor» – текстовый редактор, предназначенный для создания и редактирования текстовых файлов, содержащих описание логики проекта на языке описания устройств AHDL (Altera Hardware Description Language) или на близких к нему языках типа VHDL, «Verilog».

«Symbol Editor» – символьный редактор, позволяющий редактировать существующие символы и создавать новые. Кстати, любой

откомпилированный проект может быть свёрнут в символ, помещён в библиотеку символов и использован как элемент в любом другом проекте.

«Floorplan Editor» – редактор связей (поуровневый планировщик), который на плане расположения основных логических элементов позволяет вручную распределять выводы ПЛИС (закреплять выводы за конкретными входными и выходными сигналами) и перераспределять некоторые внутренние ресурсы ПЛИС.

Приложения «MAX+plus II Compiler»

Это приложения, входящие в пакет компилятора и предназначенные для синтеза структуры, трассировки связей, проверки корректности проекта и локализации ошибок, формирования файлов программирования или конфигурирования ПЛИС:

«Netlist Extractor» – приложение, обеспечивающее извлечение списка соединений из исходного файла представления проекта, созданного при вводе проекта.

«Database Builder» – приложение, предназначенное для построения базы данных проекта.

«Logic Synthesizer» – приложение, обеспечивающее проверку корректности проекта по формальным правилам и синтез оптимальной структуры проекта.

«Partitioner» – приложение, обеспечивающее разбиение проекта на части в тех случаях, когда ресурсов одного кристалла (микросхемы) недостаточно для реализации проекта.

«Fitter» – трассировщик внутренних связей, обеспечивающий реализацию синтезированной структуры.

«SNF Extractor» – приложение, обеспечивающее извлечение параметров проекта, необходимых для функционального моделирования и временного анализа.

Приложения для верификации проектов

«Simulator» – приложение, которое совместно с редактором временных диаграмм предназначено для функционального моделирования проекта с целью проверки правильности логики его функционирования.

«Timing Analyzer» – приложение, обеспечивающее расчет временных задержек от каждого входа до каждого логически связанного с ним выхода.

«MAX+plus II Programmer»

Наконец, для программирования или конфигурирования ПЛИС используется приложение «MAX+plus II Programmer».

Программирование и перепрограммирование микросхем, имеющих встроенную систему программирования (ISP), может осуществляться непосредственно в составе конечного изделия через специальный кабель, подключаемый либо к LPT-порту (Byte Blaster), либо к COM-порту (Bit Blaster) компьютера и технологического 10-контактного соединителя интерфейса JTAG, устанавливаемого на плате изделия. Схемы кабелей можно найти на сайте фирмы «Альтера». Если на плате изделия

устанавливается несколько ПЛИС со встроенными системами программирования, то все они могут программироваться через один технологический разъём. Для этой цели приложение «Programmer» имеет режим «Multi-Device» (к сожалению, бесплатные версии пакета этот режим не поддерживают). Схемы подключения ПЛИС к интерфейсу JTAG приводятся в документации на «Bit Blaster» и «Byte Blaster». Для программирования остальных микросхем необходимо дополнительно использовать внешний программатор, который также может подключаться к COM- или LPT-порту.

Сервисные приложения

В состав САПР, кроме того, входят три сервисных приложения:

«Design Doctor» – приложение, предназначенное для проверки корректности проекта с использованием эмпирических правил.

«Message Processor» – процессор сообщений, обеспечивающий обработку, вывод на отображение и локализацию (указание места в проекте, к которому оно относится) сообщений трёх типов: сообщений об ошибках («Error»), предупреждений («Warning») и информационных сообщений («Info»). Причину вывода того или иного сообщения можно выяснить через опцию «Help on Message» процессора сообщений. При наличии сообщений об ошибках компиляция проекта невозможна до их полного устранения. При наличии предупреждений компиляция успешно завершается, однако наличие предупреждения свидетельствует об обнаружении проблемы, которая может привести к неверной работе устройства. Поэтому все предупреждения должны быть тщательно проанализированы с использованием «Help on Message», до выяснения причин их появления и последующего устранения этих причин (или игнорирования предупреждения, что иногда бывает возможно). Информационные сообщения нужно только принимать к сведению.

«Hierarchy Display» – приложение, обеспечивающее обзор иерархической структуры проекта, который может состоять из множества составленных в различных редакторах и свёрнутых в символы проектов более низких уровней, причём число уровней не ограничивается. Основной проект (проект самого верхнего уровня) должен быть создан в графическом редакторе (если проект имеет только один уровень иерархии, то он может быть создан в любом редакторе).

4.2 Рабочие каталоги системы

Во время инсталляции пакета создаются два каталога: каталог \MAXPLUS2, который содержит все приложения и библиотеки пакета, и каталог \MAX2WORK, который содержит подкаталог \CHIPTRIP со всеми файлами учебного проекта, рассмотренного в руководстве («MAX+plus II Tutorial»), и ряд подкаталогов, используемых электронным справочником («MAX+plus II Help»). В этом же каталоге \MAX2WORK следует размещать и рабочие каталоги создаваемых проектов устройств.

Необходимость создания отдельных каталогов для каждого разрабатываемого проекта обусловлена тем, что в процессе разработки проекта системой MAX+plus II создаётся и поддерживается множество файлов, относящихся к текущему проекту. Прежде всего это файл проекта («Project File»), название которого определяет название проекта в целом. Этот файл содержит основную логику и иерархию проекта, обрабатываемую компилятором. Кроме того, создаётся ряд вспомогательных файлов, связанных с проектом, но не являющихся частью логики проекта. Большая часть вспомогательных файлов создаётся и автоматически помещается в каталог проекта в процессе ввода и компиляции проекта. Это прежде всего файлы назначений и конфигурации (.ACF), файлы отчётов (.RPT), файлы данных для функционального моделирования и временного анализа (.SNF), файлы данных для программирования (.POF) и ряд других. Названия этих файлов всегда совпадают с названием проекта. Некоторые вспомогательные файлы создаются пользователем: например, для выполнения функционального моделирования создаётся файл (.SCF), содержащий описание начальных и текущих состояний входных сигналов (входов) и перечень выходов, для которых должны быть определены выходные сигналы. Поэтому перед началом работы над новым проектом следует создать рабочий каталог проекта, при этом имя каталога можно выбирать произвольно (т.е. имя каталога может не совпадать с именем файла проекта).

4.3 Начало работы с пакетом

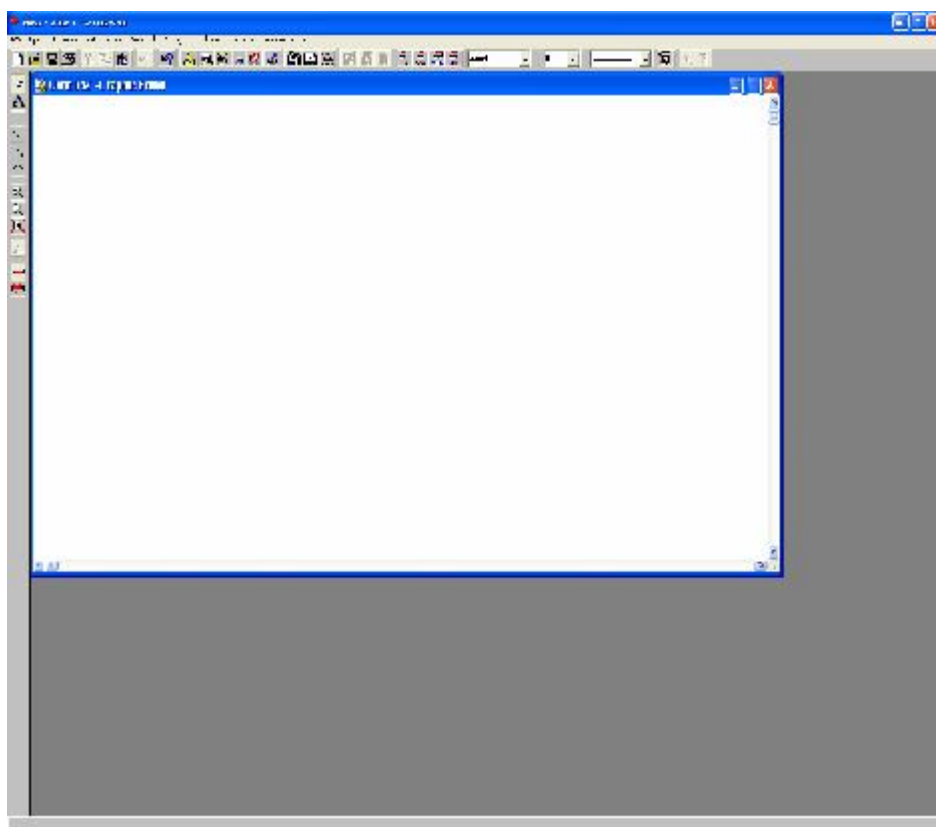


Рисунок 4.1 Окно графического редактора «Graphic Editor»

Для начала, используя стандартные средства «Виндоуз», создадим в каталоге \MAX2WORK рабочий каталог под именем \MOD_IND. Затем создадим проект схемы формирования сигнала записи в порт. Будем создавать его в графическом редакторе. Для этого запускаем систему «MAX+plus II» и создаём новый файл (крайний левый значок панели инструментов). В открывшемся диалоговом окне «New» выбираем пункт «Graphic Editor file» и нажимаем кнопку «ОК», при этом автоматически открывается окно графического редактора.

Окно редактора имеет ряд дополнительных пунктов основного меню и панель инструментов редактора, расположенную вертикально с левой стороны окна. Сохраним новый файл проекта (через меню «File > Save») под именем WRITE_PORT (расширение будет присвоено автоматически) в созданном каталоге \MOD_IND. Имя файла проекта следует обязательно привязать к имени проекта – это делается при выборе пункта «Set Project to Current File» (в подменю «Project» меню «File» главного меню рабочего окна).

Для создания графического проекта можно использовать библиотеки примитивов (\MAXPLUS\MAX2LIB\PRIM), макрофункций (\MAXPLUS\MAX2LIB\MF) и параметризованных мегафункций (\MAXPLUS\MAX2LIB\MEGA_LPM).

Примитивы включают большой набор основных логических элементов, триггеров, элементов входа и выхода (INPUT, OUTPUT, BIDIR), а также вспомогательные элементы: GND (логический ноль), VCC (логическая единица).

Макрофункции включают главным образом эквиваленты логических микросхем 74-й серии.

4.4 Работа в графическом редакторе

Прежде чем перейти непосредственно к созданию проекта схемы формирования сигнала записи в порт, рассмотрим некоторые особенности и свойства графического редактора системы «MAX+plus II».

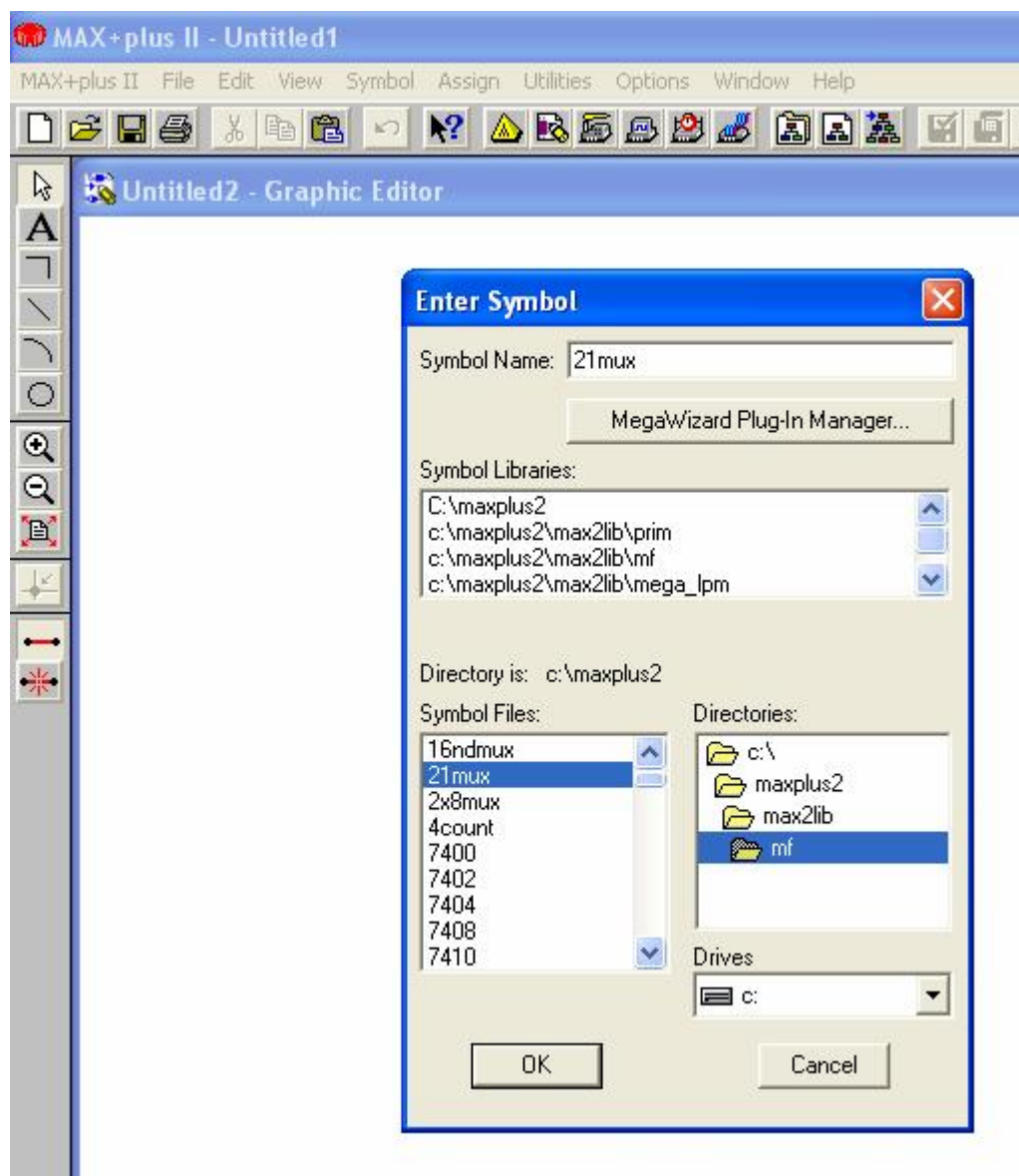


Рисунок 4.2 Диалоговое окно размещения элементов

Для размещения элементов в окне редактора используется диалоговое окно «Enter Symbol», которое можно открыть через меню «Symbol» основного меню редактора (пункт «Enter Symbol»). Однако удобнее вызывать это окно двойным щелчком левой клавиши мыши по свободному пространству открытого окна редактора, после чего выбранный элемент будет размещён именно в этом месте (позиционирование элемента производится по верхнему левому углу условного обозначения элемента). Выбрать и установить необходимый элемент можно двумя способами:

Набрать имя элемента (примитива, мега- или макрофункции) в окне «Symbol Name» диалогового окна «Enter Symbol» и нажать «OK».

Выбрать необходимую библиотеку в окне «Symbol Libraries» диалогового окна «Enter Symbol» и двойным щелчком левой клавиши мыши открыть её. Затем аналогичным образом выбрать необходимый элемент в окне «Symbol File».

Установленный элемент можно переместить в другое место обычным приёмом «Drag&Drop». Перемещение элементов может производиться без сохранения или с сохранением соединений, проведённых между элементами, в зависимости от состояния On/Off кнопок «Rubberbanding Functions», расположенных в нижней части панели инструментов редактора.

Выводы символов элементов можно соединить сигнальными линиями или присваиваемыми одинаковыми именами. Для проведения сигнальной линии необходимо совместить указатель курсора (конец стрелки) с выводом элемента, при этом курсор автоматически превращается в инструмент рисования ортогональных линий (перекрестие). После этого проводится необходимая линия (при нажатой и удерживаемой левой кнопке мыши). За один приём можно провести два ортогональных отрезка. Если этого недостаточно, то процедуру можно повторять, начиная с конца проведённой линии или с вывода другого элемента.

Для соединения элементов необязательно проводить сплошные соединительные линии между выводами. Достаточно выводам, которые должны быть соединены между собой, присвоить одинаковые имена. Поименованный вывод элемента называется узлом («Node»). Для присвоения узлу имени необходимо сначала провести короткую сигнальную линию. Так как по окончании рисования линии она остаётся выделенной (выделенные элементы обозначаются красным цветом или красной рамкой), то имя узла можно вводить сразу. Если линия не выделена (имеет чёрный цвет), то необходимо выделить её щелчком левой кнопки мыши в том месте, где должно начинаться имя узла, и набрать на клавиатуре требуемое имя.

Установленные элементы и группы элементов (любую выделенную область окна) можно удалять, перемещать, поворачивать, копировать и размножать в пределах окна редактора, а также переносить в другие окна через буфер обмена. Все эти операции можно выполнять с использованием пунктов меню «Edit» (в основном меню редактора), предварительно выделив требуемую область или элемент.

Выделение отдельных элементов, включая и соединительные линии, производится щелчком левой кнопки мыши на изображении символа элемента. Выделение области возможно при нажатой и удерживаемой левой кнопке мыши. Выделенная область обозначается рамкой чёрного цвета, а все выделенные элементы – красным цветом.

Для удаления, копирования, вырезания, поворота и некоторых других операций с элементами можно использовать выпадающее контекстное меню, которое вызывается щелчком правой кнопки мыши по выбранному элементу. Для перемещения выделенных областей в пределах окна используется способ «Drag&Drop», а для размножения выделенных областей и элементов – «Drag&Drop» при нажатой и удерживаемой клавише CTRL.

Часть инструментов редактора, расположенных на вертикальной панели с левой стороны окна, предназначена для созданий поясняющих надписей, рисунков и таблиц, которые не являются частью проекта и не обрабатываются компилятором.

Создание схемы проекта целесообразно начинать с размещения узлов входов и выходов и присвоения этим узлам имён. В нашем случае необходимо разместить входы адресных линий A0...A15, вход сигнала записи /WR (низкий активный уровень) и один выход сигнала записи в порт WR_PORT. Используем для этой цели библиотеку примитивов, которая содержит примитивы INPUT (для входов) и OUTPUT (для выходов). Каждому вводу должно быть присвоено своё имя. Для этого необходимо дважды щёлкнуть мышью на поле (входа/выхода), в появившемся чёрном прямоугольнике набрать новое имя и нажать клавишу ENTER. Переход к набору имени следующего элемента ввода может осуществляться по нажатию клавиши ENTER.

Логика разрабатываемого проекта также может быть реализована с использованием примитивов: двух элементов OR8 (8-входовый элемент «ИЛИ»), одного элемента NOR3 (3-входовый элемент «ИЛИ-НЕ») и одного инвертора NOT. В описанном выше порядке размещаем требуемые элементы и выполняем необходимые соединения.

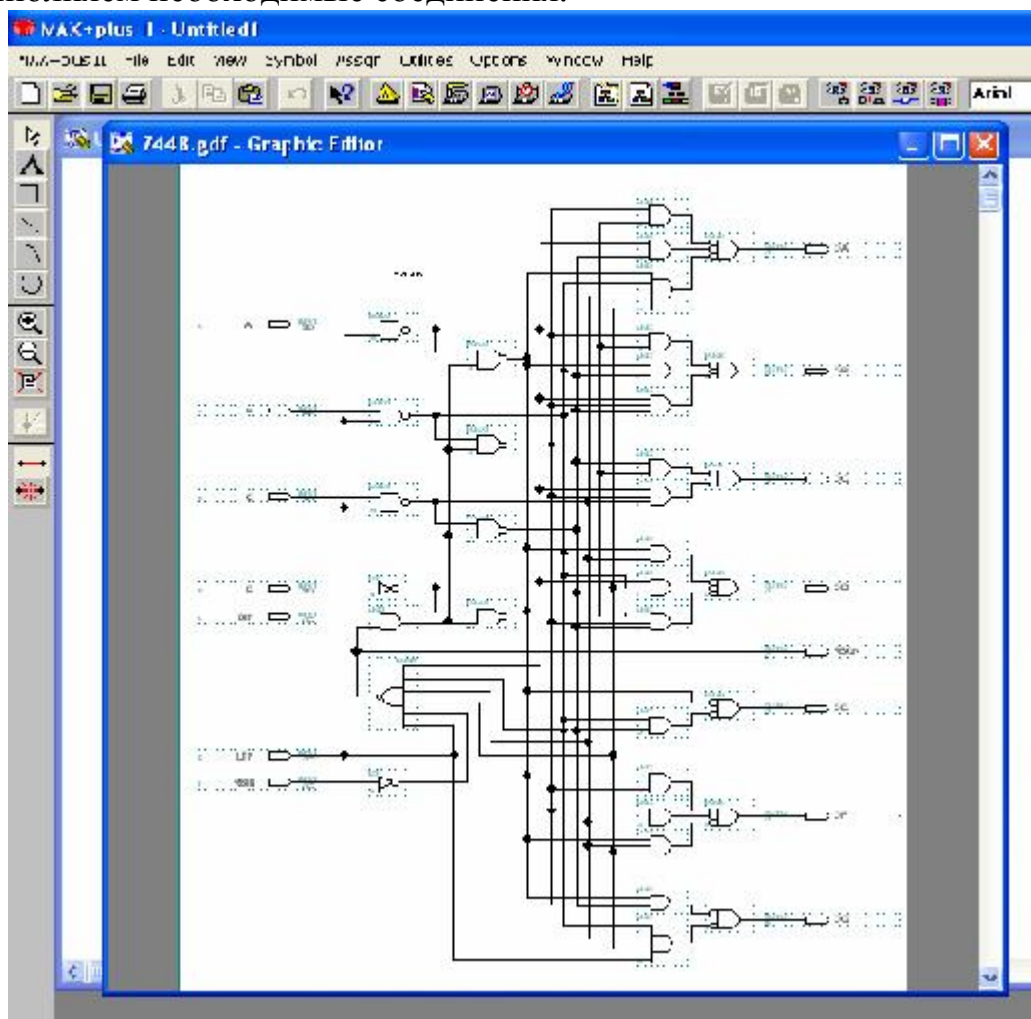


Рисунок 4.3 Создание проекта

Заметим, что предложенный вариант реализации логики устройства – не единственный и не самый лучший. Заинтересованный читатель может реализовать этот проект с использованием языка AHDL (самый предпочтительный вариант) или мегафункции LPM_OR.

Глава 5

Техника безопасности при разработке печатных плат

Так как в данном дипломном проекте рассматривается вопрос разработки устройства, сконструированного на основе печатных плат, необходимо оценить технологию производства печатных плат с точки зрения экологичности и безопасности жизнедеятельности.

В процессе производства печатных плат (при пайке) происходит выделение в атмосферу токсичных газов (оксид углерода, фтористый водород) и аэрозолей (свинец и его соединения), которые проникают в организм человека главным образом через дыхательные пути, а также через кожу и с пищей.

При пайке электропаяльником мощностью 20-60 Вт происходит выделение 0.02 - 0.04 мг вредных веществ за 100 паяк.

Большинство этих веществ относится к опасным и вредным производственным факторам, поскольку они оказывают токсическое действие на организм человека. Эти вещества, хорошо растворяясь в биологических средах, способны вступать с ними во взаимодействие, вызывая нарушение нормальной жизнедеятельности. В результате их действия у человека возникает болезненное состояние - отравление, опасность которого зависит от продолжительности воздействия, концентрации q (мг/м³) и вида вещества. По характеру воздействия на организм человека вредные вещества подразделяются на:

общетоксические - вызывающие отравление всего организма (окись углерода, цианистые соединения, свинец, ртуть, бензол, мышьяк и его соединения и др.);

раздражающие - вызывающие раздражение дыхательного тракта и слизистых оболочек (хлор, аммиак, сернистый газ, фтористый водород, окислы азота, озон, ацетон и др.);

сенсibiliзирующие - действующие как аллергены (формальдегид, различные растворители и лаки на основе нитро - и нитрозосоединений и др.);

канцерогенные - вызывающие раковые заболевания (никель и его соединения, амины, окислы хрома, асбест и др.);

мутагенные - приводящие к изменению наследственной информации (свинец, марганец, радиоактивные вещества и др.);

влияющие на репродуктивную функцию (ртуть, свинец, марганец, стирол, радиоактивные вещества и др.).

Таким образом, выделяющиеся при пайке аэрозоли свинца могут как привести к общему отравлению организма, так и повлиять на наследственную информацию и репродуктивную функцию.

Предельно допустимые концентрации (ПДК) примесей. Основной физической характеристикой примесей атмосферы является концентрация - масса (мг) вещества в единице объема (м³) воздуха при нормальных условиях. Концентрация определяет физическое, химическое и другие виды воздействий веществ на человека и окружающую среду и служат основным параметром. ПДК - это максимальная концентрация примеси в атмосфере, отнесенная к определенному времени осреднения, которая при периодическом воздействии или на протяжении всей жизни человека не оказывает ни на него ни на окружающую среду в целом вредного действия (включая отдаленные последствия). Если вещество оказывает на природу вредное действие в меньших концентрациях, чем на организм человека, то при нормировании исходят из порога действия этого вещества на окружающую природу. Устанавливается класс опасности вещества, допустимые, максимальная, разовая (ПДК_{мах}) и среднесуточная концентрация примесей (ПДК_{сс}).

Максимальная разовая ПДК_{мах} - основная характеристика опасности вредного вещества. Устанавливается для предупреждения рефлекторных реакций (ощущение запаха, световой чувствительности и т.д.) при кратковременном воздействии примесей. Среднесуточная ПДК_{сс} установлена для предупреждения общетоксического, канцерогенного, мутагенного влияния на человека.

По ГОСТ 12.1.005-76 установлены предельно допустимые концентрации вредных веществ $q_{пдк}$ (мг/м³) в воздухе рабочей зоны производственных помещений. Вредные вещества по степени воздействия на организм человека подразделяются на следующие классы: 1-й - чрезвычайно опасные, 2-й - высокоопасные, 3-й - умеренно опасные, 4-й - малоопасные.

Вещество	Величина ПДК, мг/м ³	Класс опасности	Агрегатное состояние
Свинец	0,01	1	аэрозоль

Мероприятия по оздоровлению воздушной среды

Требуемое состояние воздуха рабочей зоны может быть обеспечено выполнением определенных мероприятий, к основным из которых относятся:

1. Механизация и автоматизация производственных процессов, дистанционное управление ими. Эти мероприятия имеют большое значение для защиты от воздействия вредных веществ, теплового излучения, особенно при выполнении тяжелых работ. Автоматизация процессов, сопровождающихся выделением вредных веществ, не только повышает производительность, но и улучшает условия труда, поскольку рабочие выводятся из опасной зоны.

2. Применение технологических процессов и оборудования, исключающих образование вредных веществ или попадание их в рабочую

зону. При проектировании новых технологических процессов и оборудования необходимо добиваться исключения или резкого уменьшения выделения вредных веществ в воздух производственных помещений..

Большое значение для оздоровления воздушной среды имеет надежная герметизация оборудования, в котором находятся вредные вещества, в частности, нагревательных печей, газопроводов, насосов, компрессоров, конвейеров и т. д. Через неплотности в соединениях, а также вследствие газопроницаемости материалов происходит истечение находящихся под давлением газов. Количество вытекающего газа зависит от его физических свойств, площади неплотностей и разницы давлений снаружи и внутри оборудования.

3. Защита от источников тепловых излучений. Это важно для снижения температуры воздуха в помещении и теплового облучения работающих.

4. Устройство вентиляции и отопления, что имеет большое значение для оздоровления воздушной среды в производственных помещениях.

5. Применение средств индивидуальной защиты.

Однако даже при совершенной технологии и современном оборудовании не удастся полностью исключить попадание в воздух производственного помещения вредных веществ. В этих случаях для защиты людей используется вентиляция.

Согласно «Межотраслевым правилам по охране труда при проведении работ по пайке и лужению изделий», при проведении подобных работ с применением оловянно-свинцовых припоев необходимо соблюдать следующие требования безопасности жизнедеятельности:

при проведении работ по пайке изделий с применением оловянно - свинцовых припоев должны выполняться требования санитарных правил организации процессов пайки мелких изделий сплавами, содержащими свинец.

сборку, фиксацию, поджатие соединяемых элементов, нанесение припоя, флюса и других материалов на сборочные детали необходимо проводить с использованием специальных приспособлений или инструментов, указанных в технологической документации.

автоматы для пайки следует конструктивно оборудовать встроенными вытяжными вентиляционными устройствами.

при пайке на поточно - механизированных и автоматизированных линиях следует предусматривать местные вытяжные вентиляционные устройства, встроенные в конструкцию линий.

пайка изделий в замкнутых пространствах должна проводиться не менее чем двумя работниками. Для осуществления контроля безопасного проведения работ один из работников должен находиться вне замкнутого объема. Работник, находящийся в замкнутом пространстве, кроме спецодежды должен применять: защитные каски (полиэтиленовые, текстолитовые или винипластовые), электрозащитные средства (диэлектрические перчатки, галоши, коврики) и предохранительный пояс с

канатом, конец которого должен находиться у наблюдающего вне замкнутого объема.

пайка изделий в замкнутых пространствах должна проводиться с непрерывной работой местной приточной и вытяжной вентиляции, исключающей накопление вредных веществ в воздухе выше предельно - допустимых концентраций и поддерживающей содержание кислорода по объему не менее 19%.

при пайке интегральных микросхем должны использоваться оптические приборы, преимущественно бинокулярные стереоскопические микроскопы с телевизионными экранами.

участки пайки оловянно - свинцовыми припоями должны быть размещены в отдельных помещениях.

производственные помещения, в которых находятся участки пайки, должны быть оборудованы постоянно действующей общеобменной и местной приточно - вытяжной вентиляцией с разводкой притока в рабочую зону.

на участках, где применяются вещества первого класса опасности (оловянно - свинцовые припои, бериллий и его сплавы, флюсы и другие соединения), системы местных отсосов должны включаться до начала работ и выключаться после их окончания. Работа вентиляционных установок должна контролироваться с помощью световой и звуковой сигнализации, автоматически включающейся при остановке вентиляции.

рабочие места при пайке оловянно - свинцовыми припоями должны быть оборудованы местными вытяжными устройствами, обеспечивающими скорость движения воздуха непосредственно на месте пайки не менее 0,6 м/сек., независимо от конструкции воздухоприемников.

Вентиляция как средство защиты воздушной среды производственных помещений

Задачей вентиляции является обеспечение чистоты воздуха и заданных метеорологических условий в производственных помещениях. Вентиляция достигается удалением загрязненного или нагретого воздуха из помещения и подачей в него свежего воздуха.

По способу перемещения воздуха вентиляция бывает естественной и искусственной (механической). Возможно также сочетание естественной и искусственной вентиляции (смешанная вентиляция).

Вентиляция бывает приточной, вытяжной или приточно-вытяжной в зависимости от того, для чего служит система вентиляции, для подачи (притока) или удаления воздуха из помещения или (и) для того и другого одновременно.

По месту действия вентиляция бывает общеобменной и местной.

Действие общеобменной вентиляции основано на разбавлении загрязненного, нагретого, влажного воздуха помещения свежим воздухом до

предельно допустимых норм. Эту систему вентиляции наиболее часто применяют в случаях, когда вредные вещества, теплота, влага выделяются равномерно по всему помещению. При такой вентиляции обеспечивается поддержание необходимых параметров воздушной среды во всем объеме помещения.

Воздухообмен в помещении можно значительно сократить, если улавливать вредные вещества в местах их выделения. С этой целью технологическое оборудование, являющееся источником выделения вредных веществ, снабжают специальными устройствами, от которых производится отсос загрязненного воздуха. Такая вентиляция называется местной вытяжкой.

Местная вентиляция по сравнению с общеобменной требует значительно меньших затрат на устройство и эксплуатацию.

В производственных помещениях, в которых возможно внезапное поступление в воздух рабочей зоны больших количеств вредных паров и газов, наряду с рабочей предусматривается устройство аварийной вентиляции.

На производстве часто устраивают комбинированные системы вентиляции (общеобменную с местной, общеобменную с аварийной и т.п.).

Для эффективной работы системы вентиляции важно, чтобы еще на стадии проектирования были выполнены следующие технические и санитарно-гигиенические требования.

1. Количество приточного воздуха должно соответствовать количеству удаляемого (вытяжки); разница между ними должна быть минимальной.

В ряде случаев необходимо так организовать воздухообмен, чтобы одно количество воздуха обязательно было больше другого. Например, при проектировании вентиляции двух смежных помещений, в одном из которых выделяются вредные вещества. Количество удаляемого воздуха из этого помещения должно быть больше количества приточного воздуха, в результате чего в помещении создается небольшое разрежение.

Возможны такие схемы воздухообмена, когда во всем помещении поддерживается избыточное по отношению к атмосферному давление. Например, в цехах электровакуумного производства, для которого особенно важно отсутствие пыли.

2. Приточные и вытяжные системы в помещении должны быть правильно размещены. Свежий воздух необходимо подавать в те части помещения, где количество вредных веществ минимально, а удалять, где выделения максимальны.

Приток воздуха должен производиться, как правило, в рабочую зону, а вытяжка из верхней зоны помещения.

3. Система вентиляции не должна вызывать переохлаждения или перегрева работающих.

4. Система вентиляции не должна создавать шум на рабочих местах, превышающий предельно допустимые уровни.

5. Система вентиляции должна быть электро-, пожаро- и взрывобезопасна, проста по устройству, надежна в эксплуатации и эффективна.

Естественная вентиляция

Воздухообмен при естественной вентиляции происходит вследствие разности температур воздуха в помещении и наружного воздуха, а также в результате действия ветра.

Естественная вентиляция может быть неорганизованной и организованной.

При неорганизованной вентиляции поступление и удаление воздуха происходит через неплотности и поры наружных ограждений (инфильтрация), через окна, форточки, специальные проемы (аэрация).

Организованная естественная вентиляция осуществляется аэрацией и дефлекторами, и поддается регулировке.

Аэрация. Осуществляется в холодных цехах за счет ветрового давления, а в горячих цехах за счет совместного и отдельного действия гравитационного и ветрового давлений. В летнее время свежий воздух поступает в помещение через нижние проемы, расположенные на небольшой высоте от пола (1-1,5 м).

Поступление наружного воздуха в зимнее время осуществляется через проемы, расположенные на высоте 4-7 м от пола. Высота принимается с таким расчетом, чтобы холодный наружный воздух, опускаясь до рабочей зоны, успел достаточно нагреться за счет перемешивания с теплым воздухом помещения. Меняя положение створок, можно регулировать воздухообмен.

При обдувании зданий ветром с наветренной стороны создается повышенное давление воздуха, а на заветренной стороне разрежение.

Под напором воздуха с наветренной стороны наружный воздух будет поступать через нижние проемы и, распространяясь в нижней части здания, вытеснять более нагретый и загрязненный воздух через проемы в фанаре здания наружу. Таким образом, действие ветра усиливает воздухообмен, происходящий за счет гравитационного давления.

Преимуществом аэрации является то, что большие объемы воздуха подаются и удаляются без применения вентиляторов и воздуховодов. Система аэрации значительно дешевле механических систем вентиляции.

Недостатки: в летнее время эффективность аэрации снижается вследствие повышения температуры наружного воздуха; поступающий в помещение воздух не обрабатывается (не очищается, не охлаждается).

Вентиляция с помощью дефлекторов. Дефлекторы представляют собой специальные насадки, устанавливаемые на вытяжных воздуховодах и использующие энергию ветра. Дефлекторы применяют для удаления загрязненного или перегретого воздуха из помещений сравнительно

небольшого объема, а также для местной вентиляции, например, для вытяжки горячих газов от кузнечных горнов, печей и т.д.

Механическая вентиляция

В системах механической вентиляции движение воздуха осуществляется вентиляторами и в некоторых случаях эжекторами.

Схемы механической вентиляции:

- приточная
- вытяжная
- приточно-вытяжная

(1 – вентилятор, 2 – воздуховод, 3 – приточная насадка (щелевые отверстия), 4 – местный отсос, 5 – воздухонагреватель и увлажнитель, 6 – воздухоприемник, 7 – выпускная труба)

Вытяжная вентиляция. При работе вытяжной системы чистый воздух поступает в помещение через неплотности в ограждающих конструкциях. В ряде случаев это обстоятельство является серьезным недостатком данной системы вентиляции, так как неорганизованный приток холодного воздуха (сквозняки) может вызвать простудные заболевания.

Приточно-вытяжная вентиляция. В этой системе воздух подается в помещение приточной вентиляцией, а удаляется вытяжной вентиляцией, работающими одновременно.

Приточно-вытяжная вентиляция с рециркуляцией характерна тем, что воздух, отсасываемый из помещения вытяжной системой, частично повторно подают в это помещение через приточную систему, соединенную с вытяжной системой воздуховодом. Регулировка количества свежего, вторичного и выбрасываемого воздуха производится клапанами. В результате использования такой системы достигается экономия расходуемой теплоты на нагрев воздуха в холодное время года и на его очистку.

Для рециркуляции разрешается использовать воздух помещений, в которых отсутствуют выделения вредных веществ или выделяющиеся вещества относятся к 4-му классу опасности, причем концентрация этих веществ в подаваемом в помещение воздухе не превышает 0,3 концентрации ПДК.

Местная вентиляция

Местная вентиляция бывает приточной и вытяжной.

Местная приточная вентиляция служит для создания требуемых условий воздушной среды в ограниченной зоне производственного помещения. К установкам местной приточной вентиляции относятся: воздушные души и оазисы, воздушные и воздушно-тепловые завесы.

Воздушное душирование применяют в горячих цехах на рабочих местах под воздействием лучистого потока теплоты интенсивностью 350 Вт/м² и более. Воздушный душ представляет собой направленный на рабочего поток воздуха. Скорость обдува составляет $\bar{1} \text{ } 3,5$ м/с в зависимости от

интенсивности облучения. Эффективность душирующих агрегатов повышается при распылении воды в струе воздуха.

Воздушные оазисы - это часть производственной площади, которая отделяется со всех сторон легкими передвижными перегородками и заполняется воздухом более холодным и чистым, чем воздух помещения.

Воздушные и воздушно-тепловые завесы устраивают для защиты людей от охлаждения проникающим через ворота холодным воздухом, проникающим через ворота. Завесы бывают двух типов: воздушные с подачей воздуха без подогрева и воздушно-тепловые с подогревом подаваемого воздуха в калориферах.

Работа завес основана на том, что подаваемый воздух к воротам выходит через специальный воздуховод с щелью под определенным углом с большой скоростью (до 10-15 м/с) навстречу входящему холодному потоку и смешивается с ним. Полученная смесь более теплого воздуха поступает на рабочие места или (при недостаточном нагреве) отклоняется в сторону от них. При работе завес создается дополнительное сопротивление проходу холодного воздуха через ворота.

Местная вытяжная вентиляция. Ее применение основано на улавливании и удалении вредных веществ непосредственно у источника их образования.

Устройства местной вытяжной вентиляции делают в виде укрытий или местных отсосов.

Укрытия с отсосом характерны тем, что источник вредных выделений находится внутри них. Они могут быть выполнены как укрытия-кожухи, полностью или частично заключающие оборудование (вытяжные шкафы, витринные укрытия, кабины и камеры). Внутри укрытий создается разрежение, в результате чего вредные вещества не могут попасть в воздух помещения. Такой способ предотвращения выделения вредных веществ в помещении называется аспирацией. Аспирационные системы обычно блокируют с пусковыми устройствами технологического оборудования с тем, чтобы отсос вредных веществ производился не только в месте их выделения, но и в момент образования.

Полное укрытие машин и механизмов, выделяющих вредные вещества, наиболее совершенный и эффективный способ предотвращения их попадания в воздух помещения. Важно еще на стадии проектирования разрабатывать технологическое оборудование таким образом, чтобы такие вентиляционные устройства органически входили бы в общую конструкцию, не мешая технологическому процессу и одновременно полностью решая санитарно-гигиенические задачи.

Защитно-обеспыливающие кожухи устанавливаются на станки, на которых обработка материалов сопровождается пылевыведением и отлетанием крупных частей, которые могут нанести травму. Это шлифовальные, обдирочные, полировальные, заточные станки по металлу, деревообрабатывающие станки и др.

Вытяжные шкафы находят широкое применение при термической и гальванической обработке металлов, окраске, развеске и расфасовке сыпучих

материалов, при различных операциях, связанных с выделением вредных газов и паров.

Кабины и камеры представляют собой емкости определенного объема, внутри которых производятся работы, связанные с выделением вредных веществ (пескоструйная и дробеметная обработка, окрасочные работы и т.д.).

Вытяжные зонты (специальные устройства в виде колпаков) применяют для локализации вредных веществ, поднимающихся вверх, а именно при тепло- и влаговыведениях.

Всасывающие панели применяют в тех случаях, когда применение вытяжных зонтов недопустимо по условию попадания вредных веществ в органы дыхания работающих. Эффективным местным отсосом является панель Чернобережского, применяемая при таких операциях, как газовая сварка, пайка и т.п.

Пылегазоприемники, воронки применяются при проведении пайки и сварочных работ. Они располагаются в непосредственной близости от места пайки или сварки.

Бортовые отсосы. При травлении металлов и нанесении гальванопокрытий с открытой поверхности ванн выделяются пары кислот, щелочей, при цинковании, меднении, серебрении чрезвычайно вредный цианистый водород, при хромировании окись хрома и т.д. Для локализации этих вредных веществ используют бортовые отсосы, представляющие собой щелевидные воздуховоды шириной 40 100 мм, устанавливаемые по периферии ванн.

Принцип действия бортового отсоса состоит в том, что затягиваемый в щель воздух, двигаясь над поверхностью жидкости, увлекает с собой вредные вещества, не давая им распространиться вверх по помещению.

Для улавливания и удаления вредных выделений непосредственно с мест их образования (при пайке печатных плат) должна устраиваться местная вытяжная вентиляция. Удаление загрязненного воздуха с ее помощью осуществляется вытяжными зонтами, установленными непосредственно над поверхностью рабочего стола, или вытяжных шкафов, соединенных с трубопроводом вытяжной вентиляции.

Для расчета массы вредных веществ, образующихся при пайке необходимо знать количество припоя, расходуемого на операции пайки. Пайка осуществляется припоем ПОС-50.

относительное количество свинца в данном припое (р) - 0,5

относительное количество свинца, испаряющегося в воздух рабочей зоны при одной пайке (r) – 0,0095

расход припоя на 1 пайку (R) – 0,1 г

примем количество паяк в час (N) равным 60.

Количество выделяемого при пайке свинца рассчитываем по следующей формуле:

$$K_c = p \cdot r \cdot R \cdot N$$

$$K_c = 0,5 \cdot 0,0095 \cdot 0,1 \cdot 60 = 0,0285 \text{ г}$$

Рабочее место монтажника организовано в виде монтажного стола. Исходя из этого находим объем рабочей зоны, а именно ширина - 1,5 м; глубина - 1 м; высота, определяемая высотой потолка цеха - 4 м.

Получаем объем $V=1*1,5*4=6\text{м}^3$.

Проверим фактическую концентрацию вредных веществ в рабочей зоне:

$$K_{\text{св}}=0,0285/6=4,75 \text{ мг/м}^3$$

ПДК свинца согласно ГОСТ 12.1.007-88 составляет 0,01 мг/м, поэтому необходимо предусмотреть меры по вентиляции воздуха.

Воздушный зонт представляет собой металлический колпак, расположенный над источником вредных выделений. Сечение всасывающего отверстия колпака должно иметь форму, геометрически подобную горизонтальной проекции зеркала вредных выделений.

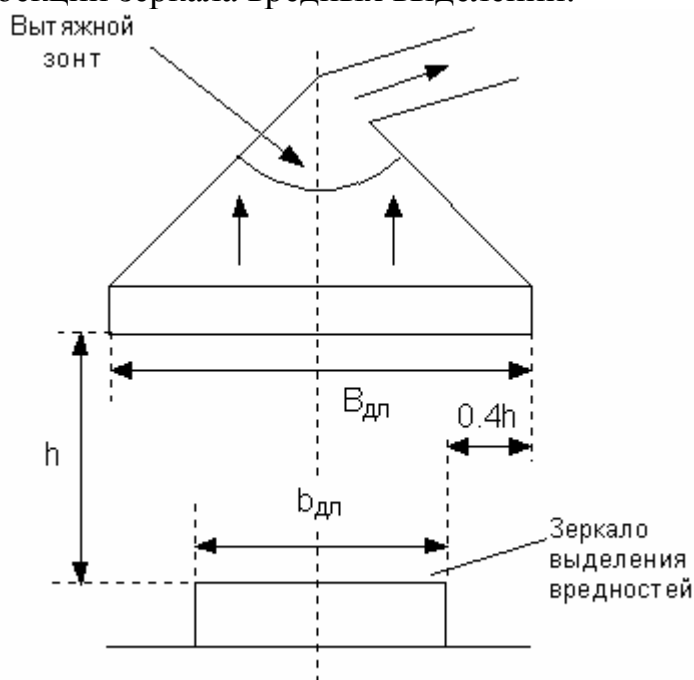


Рисунок 5.1 Зонд

Размер B каждой из сторон всасывающего сечения колпака определяется по формуле:

$$B_{\text{дл.ш}}=b_{\text{дл.ш}}+2*0,4*h, \text{ где}$$

$b_{\text{дл.ш}}$ - размер стороны (или диаметр) зеркала выделения вредностей, м,

h - расстояние от поверхности источника выделения до приемного отверстия колпака, м.

Чем меньше значение h , тем эффективнее работа зонта. Минимальное значение h определяется удобством работы при конкретном технологическом процессе. Для равномерности всасывания угол раскрытия колпака надо принимать не менее 60 градусов.

Расчет геометрических длин вытяжного зонта.

Проведем расчет сторон колпака зонта. Зеркало выделения вредностей при выполнении монтажных и сборочных работ в нашем случае имеет прямоугольную форму и не превышает следующих размеров:

длина $b_{ДЛ}$ - до 0,5 м,
 ширина $b_{Ш}$ - до 0,3 м и высота - до 0,2 м.

Расстояние h от поверхности источника выделения до приемного отверстия колпака определяется удобством работы и равно 1,5 м. Отсюда получаем, что длина и ширина колпака зонта:

$$b_{ДЛ} = b_{ДЛ} + 2 * 0,4 * b = 0,5 + 2 * 0,4 * 1,5 = 1,7 \text{ м}$$

$$b_{Ш} = b_{Ш} + 2 * 0,4 * b = 0,3 + 2 * 0,4 * 1,5 = 1,5 \text{ м}$$

Для расчета L_3 (количества воздуха, удаляемого в течение часа через зонт) необходимо определить F (площадь открытого проема зонта): $F = b_{ДЛ} * b_{Ш} = 1,7 * 1,5 = 2,55 \text{ м}^2$.

Количество воздуха, удаляемого в течение часа через зонт, $\text{м}^3/\text{ч}$, определяется по формуле:

$$L = 3600 * F * V, \text{ где}$$

F - площадь открытого проема, м^2 ;

V - средняя скорость всасывания воздуха через зонт, $\text{м}/\text{с}$

Проведем расчет L_3 , результаты запишем в таблицу.

Скорость воздуха V , $\text{м}/\text{с}$	0,1	0,2	0,3	0,4	0,5	0,6	0,7
Объем удаляемого воздуха L_3 , $\text{м}^3/\text{ч}$	918,8	1837,6	2756,4	3675,2	4594	5512,8	6431,6

Проведем расчет объема воздуха, который необходимо удалить из рабочей зоны за единицу времени:

$$L = K_c / (K_{пдк} - K_p)$$

K_c – количество паров свинца, выделяющихся в единицу времени при пайке, $\text{г}/\text{час}$,

$K_{пдк}$ – предельно допустимая концентрация паров свинца ($0,01 \text{ мг}/\text{м}^3$)

K_p – концентрация вредных примесей в воздухе, поступающих в производственное помещение (в данном расчете примем, что $K_p = 0$)

Таким образом, получаем:

$$L = 0,0285 / (0,01 * 10^{-3}) = 2850 \text{ м}^3$$

Таким образом, сравнивая полученные значения L_3 и L , можно сделать вывод, что взятый вытяжной зонт способен удалить вредные вещества, выделяющиеся в процессе производства печатных плат. Однако для этого необходимо обеспечить среднюю скорость всасывания воздуха через зонт не менее $0,4 \text{ м}/\text{с}$.

Глава 6

Технико-экономическое обоснование

В последние годы в мире получили огромное распространение Локальные Вычислительные Сети (ЛВС). Сейчас большинство организаций активно используют в своей работе удобство, которое приносит использование ЛВС. Поэтому существует необходимость оценки качества обслуживания и мониторинг работоспособности локальных сетей пользователей.

В данном дипломном проекте будут разработаны две вычислительные системы (ВС) решающие задачу оценки качества обслуживания и работоспособности ЛВС ” - одна под названием “Микропроцессорная система” - МПС, а вторая – “Контроллер на одном кристалле”.

Проектирование первой вычислительной системы - МПС осуществляется традиционными методами – на одной объединяющей плате осуществляется монтаж выбранных БИС.

Вторая вычислительная система решает те же задачи что и первая, но проектируется другими аппаратно-программными средствами, используя технологию ПЛИС – программируемые логические интегральные схемы.

Эта технология позволяет решить задачу методом “конфигурируемые системы на кристалле”. Идея весьма проста – программным путём осуществляется реконфигурация аппаратного состава выбранного кристалла, программа прошивается во Flash память, размещённую на том же кристалле, что позволяет изменять ранее прошитую программу. Новый программный продукт может быть передан в кристалл через разъём JTAG непосредственно в кристалл. Таким способом могут быть созданы на кристалле стандартные интерфейсы типа RS232, I2O (Input – Input- Output), а также хорошо известные параллельные и последовательные порты BB55 и BB51. Для удобства пользования существуют специальные библиотеки мега-функций или параметризируемых логических модулей. В конечном счете, получаются некие программы. Их трансляция осуществляется в специализированных отладочных средствах, таких как, например, пакет MAX+plus II фирмы ALTERA.

Технико-экономическое обоснование данного дипломного проекта состоит из трёх частей:

В первой рассчитываются технико-экономические показатели проектируемой микропроцессорной системы, а во второй контроллера на одном кристалле. Они включает в себя:

- определение единовременных затрат;
- расчет себестоимости изготовления и оптовой цены;
- расчет показателей надежности.

В третьей части осуществляется сравнение и оценка экономической эффективности разработанных вычислительных систем - МПС и “Контроллер на одном кристалле”.

6.1 Расчет технико-экономических показателей проектируемой микропроцессорной системы.

6.1.1 Определение единовременных затрат.

Единовременные затраты связаны с проектированием. Устройство разрабатывается инженером-программистом.

Таблица 6.1. Содержание работ и их продолжительность.

Номер этапа	Содержание этапа	Время, час
1	Изучение литературы	40
2	Подготовка материалов и справочных данных для разработки	45
3	Составление технического задания	60
4	Разработка алгоритма работы устройства	70
5	Разработка структуры программных и аппаратных средств	90
6	Разработка аппаратных средств	280
7	Разработка исходного текста программы, трансляция и отладка	110
8	Совместная отладка аппаратуры и программного обеспечения	540
	Итого:	1235

Единовременные затраты или затраты на проектирование:

$$EЗ = ЗОСН. + ЗС.С. + ЗН + ЗЛИЦ.ПО.$$

где:

ЗОСН. - основная заработная платы работников;

ЗС.С. - отчислений на социальные нужды;

ЗЛИЦ.ПО. - затрат на приобретение лицензионного программного обеспечения;

ЗН - накладных расходов.

Основная заработная плата вычисляется как произведение часовой ставки на время работы по теме:

$$ЗОСН. = 31 * ТРАБ.$$

где:

31 – часовая ставка.

ТРАБ. – время работы по теме.

Отчисления на социальные нужды исчисляются как 40,5% от фонда заработной платы и вычисляются по формуле:

$$ЗС.С.=ЗОСН.*0,405$$

Величина накладных расходов, исчисляемых пропорционально основной заработной плате работников, равна:

$$ЗН= ЗОСН*0,6$$

Затраты на приобретение лицензионного программного обеспечения, в данном случае покупки программы САПР P-CAD 2002, составят (данные найдены в поисковой системе <http://www.yandex.ru/> в Интернете на 10.03.04.):

$$ЗЛИЦ.ПО.= 299850 \text{ руб.}$$

Общее время работы по теме инженера-программиста равно 1235 часа. При часовой ставке 50 руб./час. основная заработная плата составит:

$$ЗОСН.= 1235 * 50 = 61750 \text{ руб.}$$

Отчисления на социальные нужды равны:

$$ЗС.С.= 61750 * 0,405 = 25008,75 \text{ руб.}$$

Накладные расходы будут равны:

$$ЗН = 61750 * 0,6 = 37050 \text{ руб.}$$

В итоге единовременные затраты составят:

$$ЕЗ = 61750 + 25008,75 + 37050 + 299850 = 423658,75 \text{ руб.}$$

6.1.2 Расчет себестоимости изготовления и оптовой цены.

В основе цены новой техники лежит полная себестоимость ее изготовления, которая складывается из следующих элементов затрат:

- прямые затраты (покупные изделия и полуфабрикаты, основные и вспомогательные материалы; основная заработная плата производственных рабочих);

- косвенные расходы (накладные и внепроизводственные расходы)

Полная себестоимость изготовления изделия C исчисляется по формуле:

$$C = [M + ПИ + ЗП * (1 + \alpha H)] * (1 + \beta), \text{ где}$$

M - стоимость основных и вспомогательных материалов;

$П_{и}$ - стоимость покупных изделий и полуфабрикатов;

$З_{п}$ - основная заработная плата производственных рабочих;

αH - коэффициент, учитывающий величину накладных расходов;

β - коэффициент, учитывающий величину внепроизводственных расходов.

Для определения оптовой цены изделия ЦО к себестоимости изделия прибавляется плановая рентабельность P в размере 30%:

$$Ц_o = C * \left(1 + \frac{P}{100}\right)$$

На стадии технического проектирования и подготовки производства нового оборудования его себестоимость можно определить, зная типовую структуру и величину расходов по одному из элементов прямых производственных затрат.

Руководствуясь структурной схемой устройства, составляется спецификация покупных комплектующих изделий (Таблица 2). При этом рассчитываются транспортно-заготовительные расходы в размере 10% стоимости покупных изделий.

Цены на покупные изделия взяты на основании сопоставления близких аналогов отечественных и зарубежных комплектующих изделий на 01.03.04. найденных в поисковой системе <http://www.yandex.ru/> в Интернете.

Таблица 6.2. Расчет стоимости покупных изделий и полуфабрикатов.

№	Наименование	Кол-во	Стоимость единицы, руб.	Стоимость общая, руб.
1	КР580ИР83	2	5	10
2	КР580ВА87	2	6	12
3	К1810ВГ88	2	5,5	11
4	К1810ВБ89	1	5,5	5,5
5	К1810ГФ84	1	15	15
6	К1810ВМ86	1	20	20
7	К1810ВН59А	1	45	45
8	КР541РУ2	1	25	25
9	КР580ВИ53	1	50	50
10	К155ЛА2	1	1,55	1,55
11	Кварц 5 МГц	1	10	10
Итого:		-	-	205
	Транспортно-заготовительные расходы в размере 10%, итого:	-	-	20,5
Всего:		-	-	225,55

Укрупненный расчет суммы затрат на материалы и суммы основной зарплаты проводят по следующим формулам:

$$M = \frac{P_{II}}{Y_{П.И.}} * Y_M$$

$$З_{П} = \frac{P_{II}}{Y_{П.И.}} * Y_{З.П.}$$

где:

M - стоимость основных и вспомогательных материалов;

P_{II} - стоимость покупных изделий и полуфабрикатов;

$З_{П}$ - основная заработная плата производственных рабочих;

$Y_{П.И.}, Y_M, Y_{З.П.}$ - удельный вес затрат соответственно на покупные изделия и полуфабрикаты, материалы и зарплату.

$$Y_{П.И.} = 45\%$$

$$Y_M = 10\%$$

$$Y_{З.П.} = 45\%$$

Коэффициент α_H , учитывающий величину накладных расходов при мелкосерийном производстве, исчисляемых пропорционально основной заработной плате работников, равен 2,6.

Коэффициент β , учитывающий внепроизводственные расходы, равен 1,5.

Поскольку стоимость покупных изделий и полуфабрикатов Π_{II} равна 202,35 рублей, рассчитываем:

$$M = (225,55 / 45) * 10 = 50,12 \text{ руб.}$$

$$З_{п} = (225,55 / 45) * 45 = 225,55 \text{ руб.}$$

Тогда полная себестоимость изготовления изделия С равна:

$$C = [50,12 + 225,55 + 225,55 * (1 + 2,6)] * (1 + 1,5) = 2719 \text{ руб.}$$

В конечном счете, определяется оптовая цена производства изделия:

$$ЦО = 2182 * (1 + 30/100) = 3535 \text{ руб.}$$

6.1.3 Расчет показателей надежности.

Одной из важных характеристик эксплуатационных качеств проектируемой микропроцессорной системы является ее надежность. Основными показателями надежности являются средняя наработка на отказ и коэффициент готовности.

При расчете будем считать, что все элементы схемы работают в одинаковых условиях и при номинальной нагрузке соответствуют техническим условиям. Это позволяет использовать для расчета величины интенсивности отказов элементов, записанные в технических условиях на данные элементы. Среднее время наработки на отказ определяется по формуле:

$$T_0 = \frac{1}{\sum_{i=1}^K \lambda_i n_i},$$

где

λ_i - интенсивность отказов i -го элемента;

n_i - количество i элементов (с одинаковой интенсивностью отказов);

K - количество групп элементов с одинаковой интенсивностью отказов.

Данные по интенсивности отказов различных элементов и их число приведены в Таблице 3.

Таблица 6.3. Интенсивность отказов различных элементов и их число.

Наименование и типы элементов	Число элементов, n	$\lambda_i \cdot 10^{-5}$ 1/ч	$\lambda_i \cdot 10^{-5} \cdot$ 1/ч
Микросхемы	10	0,01	0,10
Кварц	1	0,15	0,15
Кнопка	1	0,5	0,50
Итого:	-	-	0,75

Таким образом,

$$\sum_{i=1}^K \lambda_i \cdot n_i = 0.75 \cdot 10^{-5} \quad 1/\text{ч}, \text{ а}$$

$$T_0 = \frac{10^5}{0.75} = 133333 \text{ часов}$$

Коэффициент готовности Микропроцессорной системы автоматической сигнализации определяется по формуле:

$$K_{\Gamma} = \frac{T_0}{T_0 + T_B}, \text{ где}$$

T_B - среднее время восстановления работоспособности устройства, ч.

Поиск неисправности и ее устранение не представляют большой сложности, поскольку:

- все радиоэлементы схемы расположены на одной печатной плате и легко доступны для монтажа;
- сама схема обладает высокой однородностью, что облегчает поиск неисправности.

Исходя из перечисленного, принимаем $T_B = 1$ ч., тогда:

$$K_{\Gamma} = \frac{133333}{133333 + 1} = 0.999992$$

Таким образом, Микропроцессорная система имеет большой запас работоспособности.

Таблица 6.4. Результаты расчета технико-экономических показателей проектируемой микропроцессорной системы.

Показатель	Обозначение	Единицы измерения	Величина
Себестоимость	C	руб.	2719
Оптовая цена	ЦО	руб.	3535
Единовременные затраты на проектирование	ЕЗ	руб.	42365 8,75
Среднее время наработки на отказ	T_0	час.	13333 3
Коэффициент готовности	K_{Γ}	-	0,9999 92

6.2 Расчет технико-экономических показателей проектируемой вычислительной системы “Контроллер на одном кристалле”.

6.2.1 Определение единовременных затрат.

Единовременные затраты связаны с проектированием. Устройство разрабатывается инженером- программистом.

Таблица 6.5. Содержание работ и их продолжительность.

Номер этапа	Содержание этапа	Время, час.
1	Изучение литературы	35
2	Подготовка материалов и справочных данных для разработки	40
3	Составление технического задания	60
4	Разработка алгоритма работы устройства	70
5	Разработка структуры программных и аппаратных средств	90
6	Разработка аппаратных средств	130
7	Разработка исходного текста программы для контроллера, трансляция и отладка	110
8	Совместная отладка аппаратуры и программного обеспечения	280
	Итого:	815

Единовременные затраты или затраты на проектирование:

$$ЕЗ= ЗОСН.+ ЗС.С. + ЗН + ЗЛИЦ.ПО.$$

где:

ЗОСН. - основная заработная платы работников;

ЗС.С. - отчислений на социальные нужды;

ЗЛИЦ.ПО. - затрат на приобретение лицензионного программного обеспечения;

ЗН - накладных расходов.

Основная заработная плата вычисляется как произведение часовой ставки на время работы по теме:

$$ЗОСН.= 31 * ТРАБ.$$

где:

31– часовая ставка.

ТРАБ.– время работы по теме.

Отчисления на социальные нужды исчисляются как 40,5% от фонда заработной платы и вычисляются по формуле:

$$ЗС.С.=ЗОСН.*0,405$$

Величина накладных расходов, исчисляемых пропорционально основной заработной плате работников, равна:

$$ЗН= ЗОСН*0,6$$

Затраты на приобретение лицензионного программного обеспечения, в данном случае покупки программы Max+plus 2 и необходимых библиотек к ней, составят (данные найдены в поисковой системе <http://www.yandex.ru/> в Интернете на 10.03.04.):

$$ЗЛИЦ.ПО.= 240000 \text{ руб.}$$

Общее время работы по теме инженера- программиста равно 815 часов. При часовой ставке 50руб./час. его основная заработная плата составит:

$$ЗОСН.= 50 * 815 = 40750 \text{ руб.}$$

Отчисления на социальные нужды равны:

$$ЗС.С.= 40750 * 0,405 = 16503,75 \text{ руб.}$$

Накладные расходы будут равны:

$$ЗН = 40750 * 0,6 = 24450 \text{ руб.}$$

В итоге единовременные затраты составят:

$$ЕЗ = 40750 + 16503,75 + 24450 + 240000 = 321703,75 \text{ руб.}$$

6.2.2 Расчет себестоимости изготовления и оптовой цены.

В основе цены новой техники лежит полная себестоимость ее изготовления, которая складывается из следующих элементов затрат:

- прямые затраты (покупные изделия и полуфабрикаты, основные и вспомогательные материалы; основная заработная плата производственных рабочих);

- косвенные расходы (накладные и внепроизводственные расходы)

Полная себестоимость изготовления изделия С исчисляется по формуле:

$$С=[М + ПИ + ЗП * (1+\alpha Н)] * (1+\beta) , \text{ где}$$

M - стоимость основных и вспомогательных материалов;

$П_{и}$ - стоимость покупных изделий и полуфабрикатов;

$З_{п}$ - основная заработная плата производственных рабочих;

$\alpha Н$ - коэффициент, учитывающий величину накладных расходов;

β - коэффициент, учитывающий величину внепроизводственных расходов.

Для определения оптовой цены изделия ЦО к себестоимости изделия прибавляется плановая рентабельность Р в размере 30%:

$$Ц_o = C * (1 + \frac{P}{100})$$

На стадии технического проектирования и подготовки производства нового оборудования его себестоимость можно определить, зная типовую структуру и величину расходов по одному из элементов прямых производственных затрат.

Руководствуясь структурной схемой устройства, составляется спецификация покупных комплектующих изделий (Таблица 6). При этом рассчитываются транспортно-заготовительные расходы в размере 10% стоимости покупных изделий.

Цены на покупные изделия взяты на основании сопоставления близких аналогов отечественных и зарубежных комплектующих изделий на 01.03.04. найденных в поисковой системе <http://www.yandex.ru/> в Интернете.

Таблица 6.6. Расчет стоимости покупных изделий и полуфабрикатов.

№	Наименование	Ко л-во	Стоим ость единицы, руб.	Стоим ость общая, руб.
1	Мах3000 фирмы Altera	1	605,5	605,5
2	Разъем и другие расходные материалы			15
И того:		-	-	620,5
	Транспортно- заготовительные расходы в размере 10%, итого:	-	-	62,05
В сего:		-	-	682,55

Укрупненный расчет суммы затрат на материалы и суммы основной зарплаты проводят по следующим формулам:

$$M = \frac{П_{II}}{У_{П.И.}} * У_M$$

$$З_{II} = \frac{П_{II}}{У_{П.И.}} * У_{З.П.}$$

где:

M - стоимость основных и вспомогательных материалов;
 Π_{II} - стоимость покупных изделий и полуфабрикатов;
 Z_{II} - основная заработная плата производственных рабочих;
 $U_{II}, U_M, U_{Z.P.}$ - удельный вес затрат соответственно на покупные изделия и полуфабрикаты, материалы и зарплату.

$$U_{II} = 45\%$$

$$U_M = 10\%$$

$$U_{Z.P.} = 45\%$$

Коэффициент α_H , учитывающий величину накладных расходов при мелкосерийном производстве, исчисляемых пропорционально основной заработной плате работников, равен 2,6.

Коэффициент β , учитывающий внепроизводственные расходы, равен 1,5.

Поскольку стоимость покупных изделий и полуфабрикатов Π_{II} равна 682,55 рублей, рассчитываем:

$$M = (682,55 / 45) * 10 = 151,68 \text{ руб.}$$

$$Z_{II} = (682,55 / 45) * 45 = 682,55 \text{ руб.}$$

Тогда полная себестоимость изготовления изделия С равна:

$$C = [151,68 + 682,55 + 682,55 * (1 + 2,6)] * (1 + 1,5) = 8228,53 \text{ руб.}$$

В конечном счете, определяется оптовая цена производства изделия:

$$ЦО = 8228,53 * (1 + 30/100) = 10697,09 \text{ руб.}$$

6.2.3 Расчет показателей надежности.

Одной из важных характеристик эксплуатационных качеств проектируемой вычислительной системы “Контроллер на одном кристалле” является ее надежность. Основными показателями надежности являются средняя наработка на отказ и коэффициент готовности.

При расчете будем считать, что все элементы работают в одинаковых условиях и при номинальной нагрузке соответствуют техническим условиям. Это позволяет использовать для расчета величины интенсивности отказов элементов, записанные в технических условиях на данные элементы. Среднее время наработки на отказ определяется по формуле:

$$T_0 = \frac{1}{\sum_{i=1}^K \lambda_i n_i}, \text{ где}$$

λ_i - интенсивность отказов i -го элемента;

n_i - количество i элементов (с одинаковой интенсивностью отказов);

K - количество групп элементов с одинаковой интенсивностью отказов.

Данные по интенсивности отказов различных элементов и их число приведены в Таблице 7.

Таблица 6.7. Интенсивность отказов различных элементов и их число.

Наименование и типы элементов	Число элементов, n	$\lambda_i \cdot 10^{-5}$ 1/ч	$\lambda_i \cdot 10^{-5}$, 1/ч
Микросхемы	1	0,01	0,01
Разъем	1	0,3	0,3
Итого:	-	-	0,31

Таким образом,

$$\sum_{i=1}^K \lambda_i \cdot n_i = 0,31 \cdot 10^{-5} \text{ 1/ч, а}$$

$$T_0 = \frac{10^5}{0,31} = 322581 \text{ часов.}$$

Коэффициент готовности определяется по формуле:

$$K_G = \frac{T_0}{T_0 + T_B}, \text{ где}$$

T_B - среднее время восстановления работоспособности устройства, $T_B = 1$ ч., тогда:

$$K_G = \frac{322581}{322581 + 1} = 0,9999969$$

Таким образом, Микропроцессорная система имеет большой запас работоспособности.

Таблица 6.8. Результаты расчета технико-экономических показателей проектируемой вычислительной системы “Контроллер на одном кристалле”.

Показатель	Обозначение	Единицы измерения	Величина
Себестоимость	C	руб.	8228,53
Оптовая цена	ЦО	руб.	10697,09
Единовременные затраты на проектирование	ЕЗ	руб.	321703,75
Среднее время наработки на отказ	T_0	час.	322581
Коэффициент готовности	K_r	-	0,9999969

6.3 Оценка экономической эффективности разработанных вычислительных систем.

Оценка производится путем выбора наиболее эффективного варианта разработанных систем.

Таблица 6.9. Сравнительная таблица технико-экономических показателей разработанных вычислительных систем.

Показатель	МПС	Контроллер на одном кристалле
Себестоимость (руб.)	2719	8228,53
Оптовая цена (руб.)	3535	10697,09
Единовременные затраты на проектирование (руб.)	423658,75	321703,75
Среднее время наработки на отказ (час.)	133333	322581
Коэффициент готовности	0,999992	0,9999969
Время разработки (час.)	1235	815

На основе приведенных данных видно, что у вычислительной системы “Контроллер на одном кристалле” по сравнению с МПС существенно снизилась стоимость единовременных затрат на проектирование и время разработки.

При переходе на новую элементную базу – ПЛИС также улучшились такие важнейшие технические показатели как среднее время наработки на отказ и коэффициент готовности.

Список литературы

1. Современные компьютерные сети. 2-е изд. / В. Столлингс. - СПб.: Питер, 2003. - 783 с.: ил. - (Серия "Классика computer science").
2. Под ред. Ю.М. Казаринова Микропроцессорный комплект К1810. - М.: Высшая школа, 1990.
3. Лебедев О.Н. Микросхемы памяти и их применение. - М.: Радио и связь, 1990.
4. Мальцев П.П. и др. Цифровые интегральные микросхемы: Справочник. - М.: Радио и связь, 1994.
5. С.А. Майоров и др. Введение в микроЭВМ. – Л.: Машиностроение, 1988.
6. Под ред. Шахнова В.А. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник - М.: Радио и связь, 1988.
7. Калабеков Б.А. Цифровые устройства и микропроцессорные системы. — М.: Радио и связь, 2000. —336 с.
8. Угрюмов Е.П. Цифровая схемотехника.— Спб.: БХВ — Санкт-Петербург, 2000.— 528с.
9. Комолов Д.А. и др. Системы автоматизированного проектирования фирмы Altera MASX+plus II и Quartus II. — М.:ИП Радиософт, 2002 — 352с.
10. Бродин В.Б., Калинин А.В. Системы на микроконтроллерах и БИС программируемой логики— М.: Издательство ЭКОМ, 2002.— 400с.
11. Артюхов В.Г. и др. Проектирование микропроцессорной электронно-вычислительной аппаратуры:— К.: Тэхника, 1988.— 263с.
12. Воробьёв Е.П., Санин К.В. Интегральные микросхемы производства СССР и их зарубежные аналоги: Справочник. - М.: Радио и связь, 1990. - 352с.

Оглавление

Введение	5
Глава 1. Обзор существующего положения микропроцессорной техники и сетевых технологий	7
Глава 2. Анализ качества обслуживания сети пользователей	14
2.1 Основные понятия методологического аппарата	14
2.1.1 Случайные переменные	14
2.1.2 Экспоненциальное распределение	15
2.1.3 Распределение Пуассона	16
2.1.4 Нормальное распределение	17
2.1.5 Стохастические процессы	18
2.2 Анализ очередей	20
2.3 Моделирование очередей к серверам	25
Глава 3. Описание структурной схемы контроллера	40
3.1 Основные элементы	40
3.2 Интерфейс PCI	53
3.3 Разработка печатной платы	56
Глава 4. Моделирование контроллера на ПЛИС	58
4.1 Приложения системы «MAX+plus II»	58
4.2 Рабочие каталоги системы	60
4.3 Начало работы с пакетом	61
4.4 Работа в графическом редакторе	63
Глава 5. Техника безопасности при разработке печатных плат	67
Глава 6. Технико-экономическое обоснование	78
6.1 Расчет технико-экономических показателей проектируемой микропроцессорной системы	79
6.1.1 Определение единовременных затрат	79
6.1.2 Расчет себестоимости изготовления и оптовой цены	81
6.1.3 Расчет показателей надежности	84
6.2 Расчет технико-экономических показателей проектируемой вычислительной системы “Контроллер на одном кристалле”	86
6.2.1 Определение единовременных затрат	86
6.2.2 Расчет себестоимости изготовления и оптовой цены	87
6.2.3 Расчет показателей надежности	89
6.3 Оценка экономической эффективности разработанных вычислительных систем	92
Список литературы	93