Логический уровень присущ исключительно дискретным системам. 2 подуровня: переключательных схем и регистровых пересылок. Переключательные схемы подразделяются на комбинационные и последовательные; первые в отличие от последних не содержат запоминающих элементов. Используется алгебра логики, моделью конечного автомата, входными/выходными последовательностями 1 и 0. Комбинационные схемы представляются таблицей истинности, в которой каждому входному набору значений сигналов ставится в соответствие набор значений сигналов на выходах.

Последовательностные схемы могут описываться диаграммами или таблицами входов/выходов, в которых определены взаимно однозначные соответствия между входами схемы, внутренними состояниями и выходами. Подуровень регистровых пересылок характеризуется более высокой степенью абстрагирования и представляет собой описание регистров и передачу данных между ними. 2 части: информационная и управляющая.

Схемный – образуется резисторами и конденсаторами. Показателями являются напряжение и ток, представленные в функции времени или частоты. Показатели поведения: ток и …

При проектировании микро и многопроцессорных систем, содерж несколько типов микропроц наборов, необходимо решать вопросы организации памяти, взаимодействия с процессорами, организации обмена между устройствами системы и внешней средой.

**Этапы проектирования межпроцессорных систем**

1. Формализация требований к системе
2. Разработка структуры и архитектуры системы
3. Разработка и изготовление аппаратных средств и программного обеспечения системы
4. Комплексная отладка приемосдаточные испытания

Этап 1. составляются внешние спецификации, перечисляются функции системы, формализуется техническое задание на систему, формально излагаются замыслы разработчика в официальном документе.

Этап 2. Определяются функции отдельных устройств

Этап 3. После определения функций, реализуемых аппаратурой, и функций, реализуемых программами, схемотехники и программисты одно-временно приступают к разработке и изготовлению соответственно опыт-ного образца и программных средств. Разработка и изготовление 1аппаратуры состоят из разработки структурных и принципиальных схем, изготовления образца, автономной отладки. Разработка программ состоит из разработки алгоритмов, написания текста исходных программ: трансляции исходных программ в объектные программы; автономной отладки.

Этап 4. Комплексная отладка предусматривает совместные работу аппаратного и программного обеспечения МПС. На каждом этапе проектирования МПС людьми могут быть внесены неисправности и приняты неверные проектные решения. Кроме того, в аппаратуре могут возникнуть дефекты.

**Источники ошибок проектирования микропроцессоров системы**

Рассмотрим источники ошибок на первых трех этапах проектирования

Этап 1. На этом этапе источниками ошибок могут быть, логическая несогласованность требований, упущения, неточности алгоритма.

Этап 2. На данном этапе источниками ошибок могут быть: упущения функций, несогласованность протокола взаимодействия аппаратуры и программ, неверный выбор микропроцессорных наборов, неточности алгоритмов, неверная интерпретация технических требований, упущение некоторых информационных: потоков.

Этап 3. На этом этапе источниками ошибок могут быть: при разработке аппаратуры - упущения некоторых функций, неверная интерпретация технических требований, недоработка в схемах синхронизации, нарушение правил проектирования, при изготовлении прототипа - неисправности комплектующих изделий, неисправности монтажа и сборки, при разработке программных средств - упущения некоторых функций технического задания, неточности в алгоритмах, неточности кодирования.

**Процесс проектирования** – итерационный процесс. Неисправности, обнаруженные на этапе приемосдаточных испытаний, могут привести к коррекции спецификации, а, следовательно, к началу проектирования всей системы. Обнаружить неисправности необходимо как можно раньше, для этого надо контролировать корректность проекта на каждом этапе разработки.

**Проверка правильности проекта**. Основные методы контроля правильности проектирования, следующие: *верификация* – формальные методы доказательства корректности проекта; *моделирование*; *тестирование*. Существует много работ по верификации программного обеспечения, микропрограмм, аппаратуры. Однако эти работы носят теоретический характер. На практике пока используют моделирование поведения объекта и тестирование.

**Диагностика неисправности** – процесс определения причины появления ошибки по результатам тестирования. **Отладка** – процесс обнаружения ошибок и определения источников их появления по результатам тестирования при проектировании микропроцессорных систем.

**Свойство контролепригодности системы.** Успех зависит от того как спроектирована система. **Управляемость** – св-во системы, при котором её поведение поддаётся управлению, т.е. имеется возможность остановить функционирование системы в определённом состоянии и затем снова запустить её.

**Микропроцессор** — это "мозг" компьютера, который занимается обработкой данных и выполнением команд. Его основная задача — выполнять вычисления и логические операции. Чтобы понять это лучше, представь, что микропроцессор — это шеф-повар на кухне. Он не занимается сбором продуктов или мытьем посуды, он только выполняет задачи — готовит еду (вычисляет) по рецептам (командам).

**Как работает микропроцессор?**

Микропроцессор берет инструкции из памяти, выполняет их и отдает результат обратно. Эти инструкции — это обычные арифметические или логические операции (сложение, вычитание, сравнение и т.д.). Он делает это с помощью двух основных компонентов:

1. Арифметико-логическое устройство (АЛУ) — выполняет математические и логические операции.

2. Устройство управления — решает, какие операции нужно выполнять, и как координировать работу с внешними устройствами, как память или дисплей.

Чтобы все это работало, микропроцессору необходимы:

• Память (RAM и ROM) — чтобы хранить данные и программы.

• Устройства ввода-вывода — например, клавиатура для ввода или монитор для вывода.

Микропроцессор сам по себе не работает с реальными вещами напрямую, его задача — просто обрабатывать информацию. Например, он не управляет напрямую светодиодами или моторчиками.

**Что такое микроконтроллер?**

Теперь, микроконтроллер — это что-то вроде "все в одном". Это маленький компьютер, который объединяет в себе микропроцессор, память и различные периферийные устройства (например, таймеры, аналоговые входы-выходы и интерфейсы для работы с датчиками). То есть, если микропроцессор — это шеф-повар, который занимается только приготовлением, то микроконтроллер — это и шеф, и повар, и официант, который обслуживает клиента, и даже кладовщик, который следит за запасами на кухне.

Структура микроконтроллера:

1. Центральный процессор (ЦПУ) — та же часть, что и в микропроцессоре, отвечает за выполнение команд.

2. Память — встроенная память для хранения программ (обычно это флеш-память) и оперативная память (RAM) для временных данных.

3. Периферия — это то, что делает микроконтроллеры очень полезными:

* Таймеры — отсчитывают время или выполняют операции через определенные интервалы.
* Цифровые входы/выходы (GPIO) — для управления внешними устройствами, такими как светодиоды, кнопки или моторы.
* Аналого-цифровые преобразователи (ADC) — для работы с аналоговыми сигналами (например, с датчиков температуры).
* Последовательные интерфейсы (SPI, UART, I2C) — для общения с другими устройствами (например, с датчиками или памятью).

Основное различие:

• Микропроцессор — это вычислительное устройство, которое требует подключения внешних компонентов для полноценной работы (память, устройства ввода-вывода, контроллеры).

• Микроконтроллер — это полностью автономная система, которая может выполнять задачи сама, контролируя другие устройства.

Пример схемы для микропроцессора

Когда мы создаем схему с микропроцессором, мы должны подключить к нему кучу вещей:

1. Память (RAM и ROM) — для хранения данных и программ.

2. Таймеры и контроллеры — для управления временными процессами.

3. Модули ввода-вывода (I/O) — для работы с внешними устройствами.

Это похоже на организацию целого завода, где микропроцессор управляет разными цехами (внешними устройствами) и всеми процессами на предприятии.

Пример схемы для микроконтроллера

Микроконтроллер уже содержит большую часть компонентов внутри, так что внешняя схема будет проще. Обычно к микроконтроллеру подключаются:

1. Датчики — для получения данных из окружающей среды (например, температурные датчики).

2. Исполнительные устройства — такие как моторы, светодиоды и реле.

3. Питание — микроконтроллер питается от внешнего источника, например, от батарейки или через USB.

Если микропроцессор — это мозг системы, который нужно подключить к различным частям тела через нервы (шины), то микроконтроллер — это готовый "мозг с телом", который сразу готов к действию.

Пример схемы:

• Для микропроцессора: нужна внешняя память (например, DRAM), схема ввода-вывода (например, порты для клавиатуры), и коммуникационные шины (например, PCI).

• Для микроконтроллера: схема будет намного проще, нужно подключить к нему, например, датчик температуры и мотор, чтобы управлять им напрямую, используя встроенные возможности.